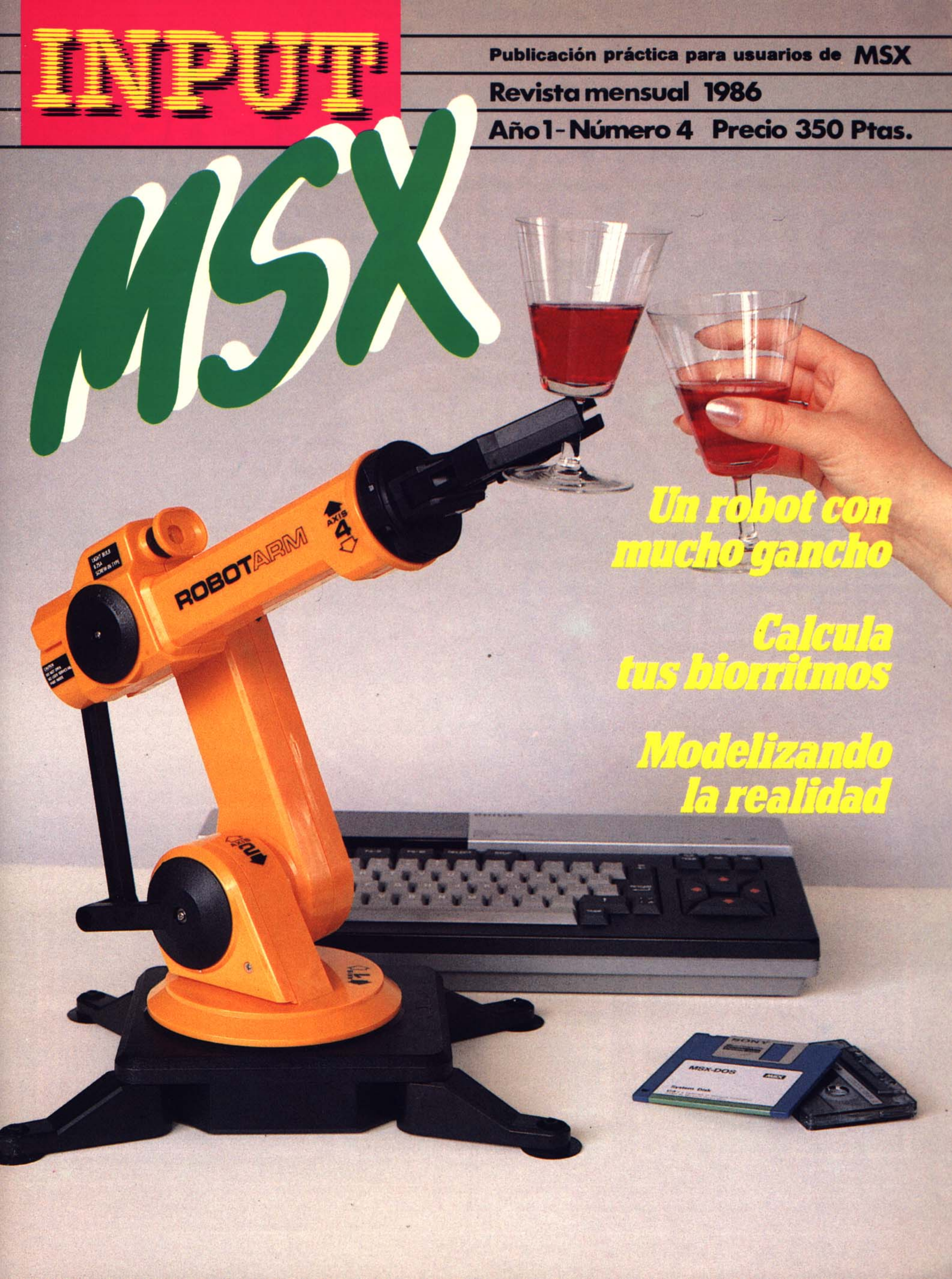

INPUT
Publicación práctica para usuarios de MSX
Revista mensual 1986
Año 1 - Número 4 Precio 350 Ptas.
MSX
Un robot con mucho gancho
Calcula tus biorritmos
Modelizando la realidad
ROBOTARM
AXIS 4
MSX-DOS
SONY



NUESTROS JUEGOS ESTAN HECHOS PARA TI

Como usuario de MSX, puede que hayas tenido la sensación de que las casas importantes de software te habían olvidado. Para remediar esa situación, ERBE, U. S. GOLD y ULTIMATE han conseguido convertir al MSX una serie de fabulosos programas pensados para aprovechar al máximo las grandes posibilidades de estos ordenadores. Aquí están cinco sobresalientes juegos en cassette que van a demostrarte lo que puede dar de sí tu MSX sin que tengas que gastar una fortuna en "cartuchos".
Directo desde Norte América llega GROG'S REVENGE. Personajes de comic inmensamente populares en U.S.A.

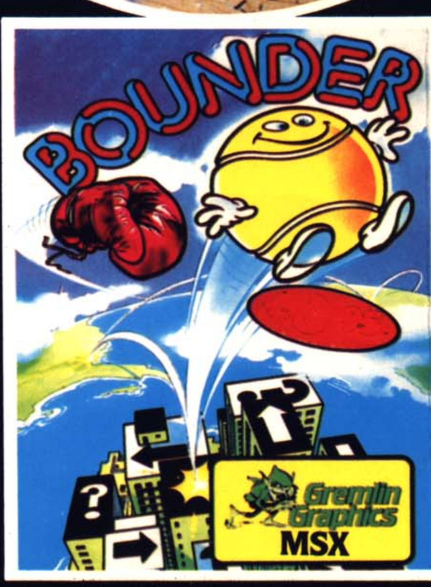


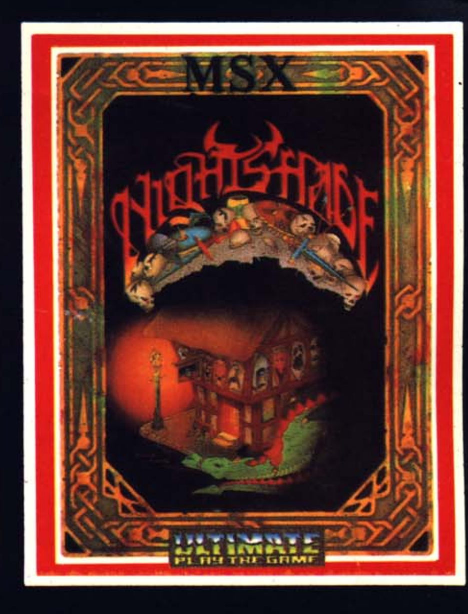
Grog y Thor son dos trogloditas, que se desenvuelven como pueden en la Edad de Piedra... y es que la supervivencia no es fácil. Diplodocus, cavernas, rocas, simas y mil peligros les acechan en el juego más original, que jamás hayas visto. "Filmation", es una técnica de programación creada y desarrollada por Ultimate para producir juegos de una calidad única. NIGHTSHADE utiliza "Filmation II" y el resultado es un juego de proporciones, que superan todo lo hecho hasta el momento en gráficos tridimensionales. BOUNDER de Gremlin Graphics es el programa que va a poner a prueba tu habilidad en el manejo del joystick. Mover una pelota de tenis que rebota a través de un sinfín de pantallas en las que has de controlar los botes para evitar que caiga donde no debe, es una tarea de lo más difícil y adictivo y que ha hecho de este juego que sea N.º 1 en Inglaterra. Y por último DAMBUSTERS. Esta simulación no necesita presentaciones, se trata de una fiel reconstrucción de la legendaria misión, que llevó a cabo durante la II Guerra Mundial el Escuadrón 617, y en la que destruyeron la presa que daba energía a las fábricas de armamento alemanas. DAMBUSTERS con su mezcla de simulador, acción y estrategia es ya un clásico dentro de los programas de ordenador. Todos estos grandes juegos en cassettes están disponibles ahora para tu MSX en todos los grandes establecimientos de informática.



BUSTERS







DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA: ERBE SOFTWARE. C/. STA. ENGRACIA, 17 28010 MADRID. TEL. (91) 447 34 10 - DELEGACION BARCELONA, AVDA. MISTRAL, N.º 10 - TEL. (93) 432 07 31



AÑO 1 NUMERO 4

DIRECTOR:
Alejandro Diges
DIRECTOR TECNICO:
Roberto Menéndez
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Antonio Taratiel, Luis R. Palencia, Francisco Tórtola, Benito Román, Esther de la Cal, Ernesto del Valle, Equipo Molisoft.

INPUT MSX es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: José Real-Grupo Jota
Madrid: c/ Gral. Varela, 35, 3.º-11
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99

FOTOMECANICA: Ochoa, S. A.

COMPOSICION: EFCA, S. A.

IMPRESION: Sirven Grafic
C/ Gran Vía, 754-756. 08013 Barcelona
Depósito legal: B-21953-1986

SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Alberto Alcocer, 46, 4.º
28016 Madrid. Teléf. 250 10 00

DISTRIBUIDORA
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona.

El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

Se ha solicitado el control OJD

INPUT MSX es independiente y no está vinculada a los distribuidores del estándar.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a través de las secciones adecuadas en estas páginas.

INPUT MSX

SUMARIO

EL UMBRAL DE LAS NUEVAS TECNOLOGIAS

Uno de los aspectos más llamativos del dinámico mercado de la informática de consumo, es el hecho de que poco a poco llegan versiones resumidas de los últimos avances tecnológicos. Sistemas de síntesis de voz, de digitalización y tratamiento de imágenes y por ejemplo, el tema de portada de este mes; los *robots*.

El bajo precio actual de estos simpáticos mecanismos se justifican las firmas, que pretenden popularizar al máximo sus lanzamientos haciendo más rentables las líneas de fabricación.

Pero lo importante es lo que puede llegar a significar esta irrupción de elementos relacionados con la alta tecnología en los hogares. Un *robot* que se mueva por el suelo buscando un enchufe para alimentarse o un brazo *robot* que lo más pesado que puede levantar es un par de ejemplares de **INPUT** no son elementos que tengan una utilidad inmediata. Su utilidad es, a más largo plazo, a través del conjunto de interrogantes que son capaces de plantear en aquellos que mantengan despierta la curiosidad.

Cambiando de tema, queremos llamar la atención sobre otro de los artículos de este mes: el programa para calcular biorritmos. Nos da pie para salir al paso de los que consideran que microordenador, como los **MSX,** no sirve más que para jugar. Nada más lejos de la realidad. No sólo hay actualmente muchas aplicaciones interesantes alejadas del ámbito de los juegos, sino que además, las aplicaciones futuras son ilimitadas. Muchas de ellas no han sido siquiera imaginadas por el más fecundo de los escritores de ciencia ficción.

Hay un gran camino que recorrer todavía en esta dirección. Ahora mismo el *hardware* es lo bastante potente como para soportar aplicaciones mucho más complejas que las actuales. Ahí mismo tenemos por ejemplo a los **MSX2.** Lo que hace falta es un esfuerzo de imaginación y sobre todo ganas de programar para dar forma a las futuras aplicaciones.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.º B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido ____________________

Segundo título elegido ____________________

Tercer título elegido ____________________

Programa que te gustaría conseguir ____________________

Qué ordenador tienes ____________________

Nombre ____________________

1.er Apellido ____________________

2.º Apellido ____________________

Fecha de nacimiento ____________________

Teléfono ____________________

Dirección ____________________

Localidad ____________________

Provincia ____________________



KONAMI VA DE TIENDAS

La conocida firma japonesa Konami, creadora de algunos de los mejores títulos de *software* de entretenimiento, parece decidida a abrir tienda en nuestro país.
Esta iniciativa es fruto del acuerdo que firmaron Konami y Serma el pasado mes de febrero (del que nos hicimos eco en esta misma sección).
Un portavoz de la firma española, que se muestra cada vez mas satisfecha con la distrubución del *software* japonés, nos comentó como nota simpática: "factores decisivos en la negociación parecen haber sido los toros y el buen vino español".
Lo cierto es que los nipones estan dispuestos a volcarse en esta interesante iniciativa.
Segun Serma, la tienda se abrirá en Madrid, probablemente durante el mes de julio y ofrecera todo el catálogo de Konami.
Ademas de los títulos clásicos habrá novedades interesantes, entre ellas la versión para MSX de Green Beret y los nuevos títulos Sao Lin Road, Kung Fu II y Nemesis, que apareceran a la vez en versiones para Spectrum, Commodore, Amstrad y por supuesto MSX.



MSX O PC



Parece que Spectravideo quiere salvar las distancias que separan a los MSX de los ordenadores tipo PC (IBM y compatibles).
No hay más que echar un vistazo a los nuevos títulos MSX: Multiplan, Dbase II, Supercalc, Wordstar y los lenguajes Turbo Pascal, FORTRAN, COBOL y C.
Ademas del *software*, Spectravideo comercializa la unidad de *diskettes* de 5 1/4 pulgadas SVI-707 (en la foto) que se conecta a cualquier MSX. Con ella es posible leer ficheros ASCII grabados en un PC y, a la inversa, grabar ficheros ASCII que podrá leer un compatible.
Spectravideo, que además lanza un compatible PC a bajo precio (unas 150.000 pesetas) pretende acercar a los usuarios de MSX las posibilidades de los compatibles.



ORDENADOR FABRICADO EN ESPAÑA



No es nuestra costumbre mencionar en esta sección equipos de otras marcas.

A no ser que vengan precedidos por algún mérito especial.
Este es el caso del compatible PC que fabrica Promax y comercializa Multihard.
Promax es una empresa fabricante de instrumentos electrónicos de medida, que goza de excelente reputación mas allá de nuestras fronteras.
Confiamos en que este sea el principio de un fructifero camino que lleve a nuevas iniciativas de fabricación.

LOS DIEZ GANADORES DEL CONCURSO INPUT/ANAYA MULTIMEDIA

A continuación mostramos la lista de los diez elegidos entre quienes habeis respodido a la primera convocatoria del concurso.

* Jesus Montero Hernando. Talavera de la Reina. Toledo.
* Juan Leiva Delgado. Linares. Jaen.
* Ricardo Dusil Munoz. Madrid.
* Jesus Barbany Gonzalez. Las Palmas de Gran Canaria.
* Luis Maria Ramos. Renteria. Guipuzcoa.
* Raul Ayllon De Miguel. Calahorra. La Rioja.
* Antonio Ferreiro Valbuena. Madrid.
* Gullermo las Heras. Vitoria.
* Isisdro Vicente Sanchez. Barcelona.
* Miguel Angel Fernadez Gutierrez. Aviles. Oviedo.

En breve plazo recibiréis una carta notificandoos el premio, junto con el catálogo de títulos de la editorial para que elijais los deseados. Gracias por vuestra participación.

DALE LA MANO AL ROBOT



Próximos están ya los tiempos en los que todo hogar contará con un miembro más. Su nombre será Robo, R2D2 **o algo similar. Será un trabajador incansable, de vivos colores y seguramente de plástico. Nos estamos refiriendo al robot doméstico.**

Una primera versión, simplificada y parcial, de estos futuros amigos la tenemos en el SVI-2000 (Robotarm) **de** Spectravídeo; **un simpático brazo robot con el que hemos tenido ocasión de trabajar.**

Tras abrir la caja en la que llegó embalado **Robotarm** (léase «el brazo robot») nos encontramos frente a frente con un simpático brazo, de sólido aspecto, constituido por una plataforma o base, que soporta las partes que constituyen el brazo, el antebrazo y la mano. Todo el conjunto, de plástico y en colores amarillo y negro, nos recordó, por su aspecto resistente, a alguno de los brazos robot que actualmente utiliza la industria, en labores de soldadura, de traslado de piezas, etc. Además del brazo en sí, nos encontramos con un conjunto de piezas sueltas, en plástico negro, que pronto identificamos como las patas y el juego de pinzas.



Las patas, que son cuatro, encajan en unos resaltes de la base de **Robotarm**. En los extremos llevan unas ventosas que se adhieren fuertemente a la mesa sobre la que descansa el brazo, permitiendo que este se mueva incluso transportando cargas pesadas, sin que en ningún momento se llegue a la pérdida del equilibrio que daría con el brazo en el suelo. A esta gran adherencia contribuye el hecho de que, en la base del robot se alojan las 4 pilas de 1.5 voltios que alimentan el conjunto. Es una verdadera lástima que no se haya previsto la posibilidad de alimentar el robot a través de un transformador. Hay que utilizar necesariamente las pilas, lo que a la larga resulta mucho más caro.

Las ventosas y el peso de las pilas aseguran la estabilidad del brazo en las condiciones normales de operación. Por su parte el juego de pinzas está formado por dos pinzas clásicas, de las de abrir y cerrar, por una especie de pala de excavadora pensada seguramente para materiales como la arena, harina o similares, que todos los chavales se empeñarán en manejar con el robot, y por último, por un soporte de plástico que encierra un imán destinado al trabajo con piezas metálicas. Es una pena que el imán sea de tan poco poder atractivo. Lo más pesado que pudimos levantar con el, fueron dos «clips» de los de sujetar papeles. Para que podaís juzgar por vosotros mismos, hemos preparado unas fotos de cada uno de los tres tipos de pinzas.

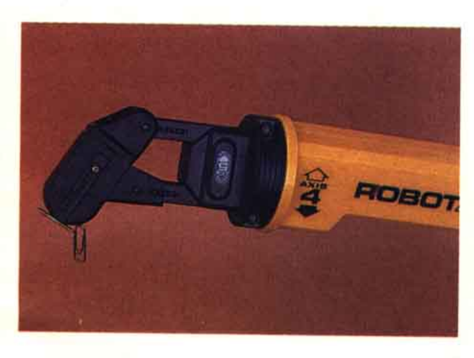
El *interface* que permite conectar el brazo a cualquier ordenador **MSX**, no iba a llegarnos hasta dos días después, y sin embargo teníamos unas ganas locas de ver a **Robotarm** funcionando. ¿Que hicimos? Muy sencillo. Este robot está pensado para funcionar de

forma autónoma, sin necesidad de estar conectado a ningún ordenador. Lo único que hace falta es conectar un par de *joysticks* en los conectores tipo **D** que hay en la base del robot. Una vez conectados, basta con accionar los *joysticks* para que se mueva el robot.

Así que pusimos las pilas, conectamos los *joysticks* y empezamos a jugar con **Robotarm**.



La libertad de movimientos de todo robot está determinada por el número de ejes alrededor de los que puede girar. Cada giro proporciona lo que se conoce como un grado de libertad. **Robotarm** es capaz de girar, de forma independiente, alrededor de 4 ejes y además es capaz de abrir y cerrar las pinzas. Esto nos da un total de 5 grados de libertad. Los ejes de giro están: uno en la base, otro al principio del brazo (lo que correspondería al hombro), el tercero entre el brazo y el antebrazo (en el codo) y el último entre el antebrazo y la mano (en la muñeca). Se hecha de menos otro eje de giro en la muñeca, perpendicular al que hay, que permitiera doblar la muñeca hacia arriba y hacia abajo. Con el único eje incorporado, la muñeca sólo puede girar. El giro alrededor de los ejes es limitado. La base tiene un recorrido total de 270 grados, el brazo 90, el antebrazo 85 y la muñeca un máximo de 108 grados.



Manejar el robot mediante los dos *joysticks* resulta sumamente sencillo. Cada *joystick* permite controlar el giro de dos ejes, ya tenemos los cuatro ejes, y además, con el botón de disparo de un *joystick* se abren las pinzas, mientras que con el del otro se cierran. De esta forma, se controlan los 5 grados de libertad.

Para probar el brazo estuvimos cogiendo y transportando de un sitio a otro de la mesa todo tipo de objetos: *cassettes*, *diskettes*, manuales de usuario finos y gruesos, cajas de plástico y cartón, destornilladores, etc. El brazo respondió bien. Se puede considerar que es un poco lento en sus movimientos y, desde luego no está pensado para levantar demasiado peso. Pero nos resultó muy entretenido trabajar con él, además de muy instructivo. Enseguida se nos plantearon algunas interesantes cuestiones que tienen que ver con la robótica, por ejemplo cómo conseguir, con la combinación simultánea de varios giros, un tipo de movimiento determinado, o qué camino recorrer y de que forma para ir de un sitio a otro en el menor tiempo posible.



Estas y otras cuestiones parecidas son las que uno se plantéa al trabajar con un robot.

EL INTERFACE Y ROGO

Desde luego más interesante que manejar el robot utilizando un par de *joysticks*, es hacer la conexión al ordenador, a través del cartucho de *interface* **Spectravídeo SVI-2000C** y controlar los movimientos del robot directamente desde un programa, haciendo uso del lenguaje de programación, incorporado en el cartucho, que ha sido bautizado con el nombre de **ROGO**.

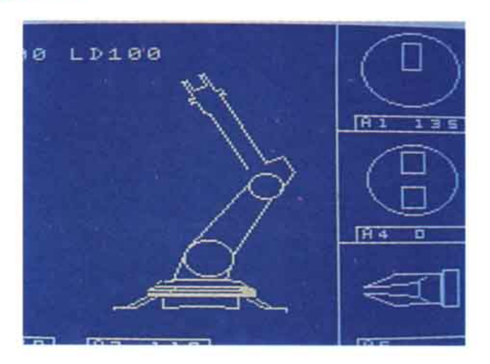
La conexión del robot al ordenador es bastante sencilla. En primer lugar

```
PRIMITIVAS DEL LENGUAJE ROGO

Comandos de manejo del robot
BC        BA        LU        LD        FU        FD        WC
WA        FO        FC
Para el robot grafico simulado
SHOWARM   HIDEARM   SHOWTEXT  HIDETEXT  CLS
Comandos para entrada/salida
PRINT     LOCATE    READC     READL     LOAD      SAVE      J1
J2
Para el control de flujo
TEST      IFTRUE    IFFALSE   REPEAT    STOP      WAIT      IF
OP        RUN
De tratamiento de listas y palabras
FIRST     WORD      SE        LPUT      NUMBERP   WORDP     BF
LISTP     EMPTYP
Para operaciones aritmeticas
SUM       DIFF      PROD      QUOT      INT       ROUND
Para operaciones logicas
<         >         =         AND       NOT       XOR       OR
Variables
MAKE      THING
```

se coloca el cartucho de *interface* en el conector de cartuchos del ordenador. Luego, mediante un par de cables que vienen incluidos con el *interface* y que tienen conectores tipo D, como los de un *joystick*, en ambos extremos, se conectan los dos conectores del lado derecho del cartucho con los dos conectores que hay en la base del robot. Ya sólo queda pulsar el interruptor de encendido del ordenador y esperar a que aparezca en la pantalla el mensaje de bienvenida del lenguaje **ROGO**.

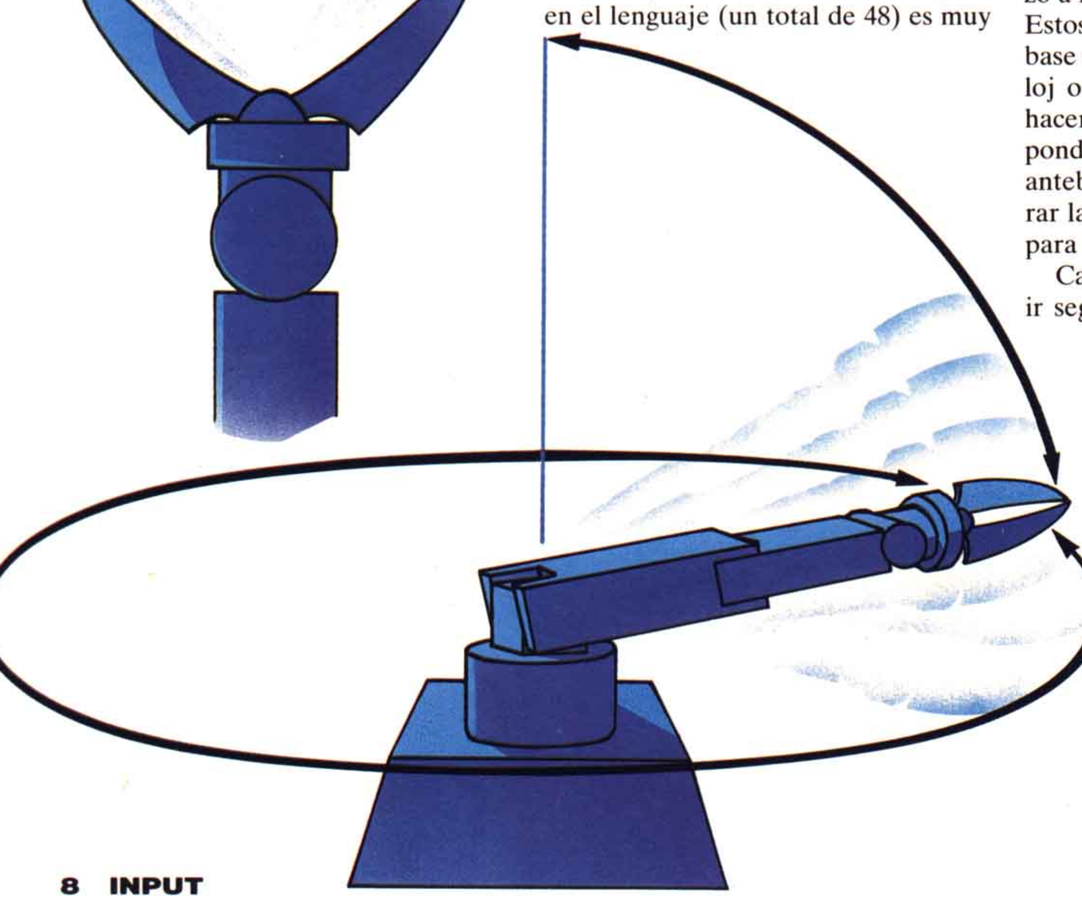
Tras este mensaje aparece el símbolo **?**, que es el cursor de **ROGO** y que nos indica que el intérprete de comandos de este lenguaje está preparado para recibir nuestras instrucciones. **ROGO** es todo un lenguaje de programación, similar en muchos aspectos al famoso **LOGO**. Al menos también permite la creación de nuevos comandos definibles por el usuario y es similar, también, en cuanto a la construcción de programas y elaboración de procedimientos.



La característica más interesante de **ROGO** es que ha sustituido los comandos de manejo de la tortuga del **LOGO**, por un conjunto de comandos para el manejo del robot.

Además del robot físico, **ROGO** es capaz de manejar una imagen simulada del mismo, presentada en la pantalla de alta resolución, que responde a los comandos exactamente de la misma forma en que lo hará el robot. Esta facilidad va a permitir, por ejemplo, observar en la pantalla los resultados de un programa de movimiento, antes de pasar al robot real, lo que puede ser de gran ayuda a la hora de depurar programas.

El conjunto de comandos incluidos en el lenguaje (un total de 48) es muy completo. No sólo hay comandos para el manjeo del robot, sino también para el almacenamiento y carga de programas en *cassette*, para realizar todo tipo de operaciones aritméticas (SUM, DIFF, etc.), lógicas (AND, OR, etc.), de control de flujo de los programas (IF, IFTRUE, REPEAT, etc.), de gestión de la entrada y salida de datos (PRINT, READ, etc) de manejo de palabras y listas de palabras (FIRST, WORD, LISTP, etc.) y de manejo y control de la imágen gráfica del robot (SHOWARM, HIDEARM, etc.). Utilizando estos comandos, unas cuantas variables, y construyendo procedimientos y subprogramas, es posible hacer muchas cosas interesantes en lenguaje **ROGO**, y no sólo relacionadas con el movimiento del robot.



ESCRIBIENDO UN PROGRAMA EN ROGO

Para empezar a escribir programas en **ROGO** con los que mover el robot, no hace falta más que echar un vistazo a los 10 comandos de movimientos. Estos son BC y BA, para mover la base en el sentido de las agujas del reloj o en el contrario, LU y LD para hacer que suba o baje el tramo correspondiente al brazo, FU y FD para el antebrazo, WC y WA para hacer girar la muñeca y, por último, FO y FC para abrir o cerrar las pinzas.

Cada uno de estos comandos debe ir seguido de un espacio y de un número que representa cuantas unidades de tiempo durará el movimiento. Esto de las unidades de tiempo no es más que una forma de medir el ángulo de giro alrededor de los ejes. Cuanto mayor sea el tiempo, mayor será el ángulo girado. Por ejemplo:

```
BC 100 (ENTER)
```

hará que la base gire en el sentido de las agujas del reloj durante 100 unidades de tiempo, mientras que:

```
FU 250 (ENTER)
```

obligará a que la parte del antebrazo se mueva hacia arriba durante 250 unidades de tiempo.

Estos comandos se ejecutan de forma directa al pulsar (ENTER). De esta forma se pueden ejecutar líneas de comandos de hasta 256 caracteres incluyendo, en la misma línea, comandos para distintos movimientos. Todos ellos se ejecutarán de forma simultánea. Por ejemplo la siguiente línea de comandos:

```
BC 100 LU 100 FU 100 WC 100
```

hará que, de forma simultánea y durante 100 unidades de tiempo, giren la base y la muñeca y se eleven el brazo y el antebrazo.

Además, resulta muy fácil definir nuevos comandos en los que se combinen varios movimientos y que no se ejecutarán hasta que lo decida el programador. Para ello hay que teclear:

```
TO"Nombre del comando (ENTER)
```

con ello entraremos en el editor de comandos. Si ahora escribimos, por ejemplo:

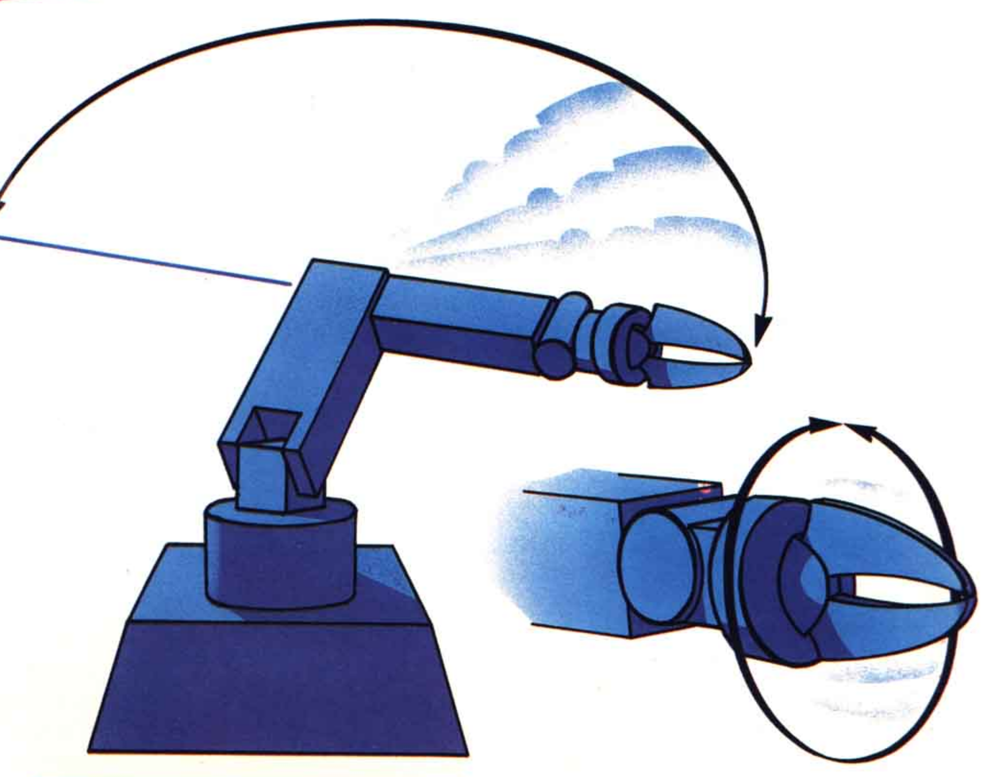
```
BC 100 LU 100
WC 200
FO 50
END
```

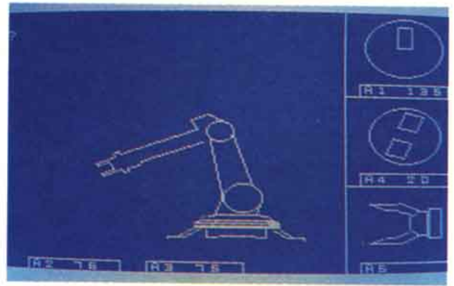
y a continuación pulsamos F5, habremos definido un nuevo comando constituido por la anterior secuencia de movimientos. Hecho esto, cada vez que tecleemos en modo directo o cada vez que aparezcan en un programa...

```
Nombre del comando
```

veremos cómo se ejecuta paso a paso la secuencia completa de movimientos.

La definición de nuevos comandos es sólo una de las muchas posibilidades que ofrece **ROGO**, que como hemos mencionado es un potente y completo lenguaje de programación. También es muy completo, afortunadamente, el manual de instrucciones que acompaña al lenguaje. Tiene cerca de 150 páginas, llenas de ejemplos, cuadros resumen y explicaciones de los diversos comandos.





¿SOLO UN JUGUETE?

Partiendo de la base de que **Robotarm** no sirve en absoluto para fabricar coches, pensemos por un momento cuál puede ser su utilidad.

Según **Spectravídeo**, el robot va dirigido a los más jovenes, y está pensado para que éstos, con su natural curiosidad, investiguen y lleguen a aprender algo escribiendo programas en **ROGO** y controlando los movimientos del robot.

Así, el robot hay que considerarlo sobre todo desde un punto de vista educativo o formativo. Es un estupendo juguete pedagógico, capaz de realizar tareas sencillas, fácil de manejar y que puede proporcionar unos buenos conocimientos sobre la interconexión de periféricos a un ordenador.

De todas formas, aunque uno no sea de los más jóvenes, es seguro que no se va a aburrir trasteando con **Robotarm** (nosotros nos lo hemos pasado en grande) y aunque las aplicaciones concretas que se le pueden dar a este robot no son evidentes, nos imaginamos que para alguno de vosotros si lo serán, y que al leer estas líneas estará diciendo: ¡Esto es justo lo que estaba buscando!

Si eres una de estas personas y ya tienes pensada una aplicación para el robot, no dudes en escribirnos contándonos el cómo, el dónde y el para qué.

NOTA

El robot y el cartucho de *interface* se comercializan de forma independiente. El precio del robot se sitúa en torno a las 11.000 pesetas, mientras que el del cartucho ronda las 8.000. **Spectravídeo** nos ha comentado que existe una versión del robot en **Kit,** pero no es probable que la misma llegue a comercializarse en España, al menos por el momento.

EL MANEJO DE LA VRAM EN SCREEN 2

- LAS TABLAS DE SCREEN 2
- COLOCANDO PUNTOS
- EL COLOR
- REPETICION DE PATRONES
- COPIANDO LA VRAM EN RAM

El BASIC MSX **te permite trabajar de una forma muy sencilla, con el modo gráfico de alta resolución. Pero si lo que quieres es aprovechar al máximo todas las posibilidades de este modo, lo mejor es que aprendas a manejar directamente la memoria de pantalla, la famosa VRAM.**

Además de la memoria RAM del sistema **MSX**, los ordenadores de este estandar poseen otro bloque de memoria RAM (la VRAM) que se ocu-



pa **exclusivamente** de contener los datos necesarios para que la presentación en pantalla sea exactamente la que tiene que ser. Esta VRAM o RAM de vídeo ocupa un total de 16 Kbytes, razón por la que algunos fabricantes indican en su publicidad, refiriéndose a los ordenadores **MSX** de 64K, que tienen 80K de RAM (64K de RAM y 16K VRAM).

El BASIC **MSX** nos permite acceder a esa memoria y leer o escribir en ella exactamente igual que en RAM. Vamos a explicar cómo está distribuida la RAM de vídeo en el modo de gráficos de alta resolución (SCREEN 2).

Para poder acceder a la VRAM necesitamos saber cómo está distribuida dicha memoria. Esto es sencillo gracias a que el BASIC **MSX** posee una variable especial que contiene las direcciones de comienzo de cada uno de los bloques de la VRAM. Esta es la variable reservada BASE(n), en la que n puede valer 0, 2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18 ó 19. Los valores de n iguales a 10, 11, 12, 13 y 14 corresponden al modo de pantalla (SCREEN 2).

COLOCANDO PUNTOS EN LA VRAM

Podemos imaginar la pantalla gráfica de alta resolución como una cuadrícula compuesta por 32 divisiones horizontales por 24 divisiones verticales (fig. 1). Numeremos esta cuadrícula asignando a cada cuadro dos números (x,y). El primero de ellos (x) variará entre 0 y 31 y el segundo (y) entre 0 y 23.

Si hacemos una ampliación del cuadro (i,j) vemos (fig. 2a) que está compuesto por 8 rebanadas horizontales, que enumeraremos de arriba hacia abajo y de 0 a 7. A su vez (fig. 2b) cada rebanada está formada por 8 puntos que enumeraremos de derecha a izquierda y de 0 a 7.

En el bloque correspondiente de VRAM, cada una de estas rebanadas se corresponde con un byte u octeto, y cada bit de ese octeto se corresponde con un punto o *pixel* de la pantalla. De esta forma, la memoria de vídeo está organizada de manera que cada grupo horizontal de 8 *pixels* está identificado por un byte.

En definitiva tenemos 32*24=768 cuadros en la cuadrícula, y al haber 8 bytes por cuadro, hacen falta 768*8=6144 bytes de memoria para almacenar toda la pantalla gráfica en VRAM.

Al dar la orden SCREEN 2 al ordenador, todos los bytes de este bloque de memoria toman el valor 0. Si situamos un punto en la pantalla, en el byte correspondiente se habrá escrito un 1 en el bit que corresponde a ese *pixel*. Si, por ejemplo, queremos situar un punto en las coordenadas (100,100) (lo que en BASIC se escribe PSET (100,100)), es decir en el bit 3 del byte número 4 del cuadro (11,11), en ese bit se escribirá un 1 (fig.3). Así un 0 quiere decir que no hay punto, con lo que el *pixel* estará del color del fondo, mientras que un 1 significa que hay punto, con lo que el *pixel* correspondiente estará del color de la tinta.

Podemos saber cuál es la dirección de comienzo que nos permite situar puntos o grupos de puntos en la pantalla, por medio de la variable BASE(12) que es igual a 0. De esta forma, si empezamos por la esquina superior izquierda (cuadro (0,0)), la rebanada 0 corresponde al byte 0 de la memoria VRAM, la rebanada 1 al byte 1, ... y la rebanada 7 al byte 7. La rebanada 0 del cuadro (1,0) (a la derecha del (0,0)) se corresponde con el byte 8 de la memoria... y la rebanada 7 de dicho cuadro, con el byte 15. El byte 16 contiene la información de la rebanada 0 del cuadro (2,0) (a la derecha del (1,0)), y así sucesivamente. Una vez completada la primera fila (cuadro (31,0)), pasamos a la segunda (cuadros (0,1) al (31,1)). Al cuadro (0,1) le corresponden los bytes 256 al 263... Y al cuadro (31,23) los bytes 6137 al 6144. Conocer esta correspondencia entre bits de la memoria de vídeo y puntos de la pantalla,

permite entre otras cosas, crear gráficos trabajando directamente sobre la memoria. Por ejemplo, mediante rutinas en lenguaje máquina. Este es el procedimiento que utilizan muchos programas comerciales y es el que permite obtener esas vertiginosas velocidades en la creación de gráficos.

Para saber cuál es el byte correspondiente a la rebanada k del cuadro (i,j), podemos utilizar la expresión:

```
BYTE=256*j+8*i+k
```

Así, el byte correspondiente a la rebanada 5 del cuadro (10,10) es el

```
256*10+8*10+5=2645
```

y el byte correspondiente a la rebanada 7 (última) del cuadro (31,23) (esquina inferior derecha) es el

```
256*23+8*31+7=6143
```

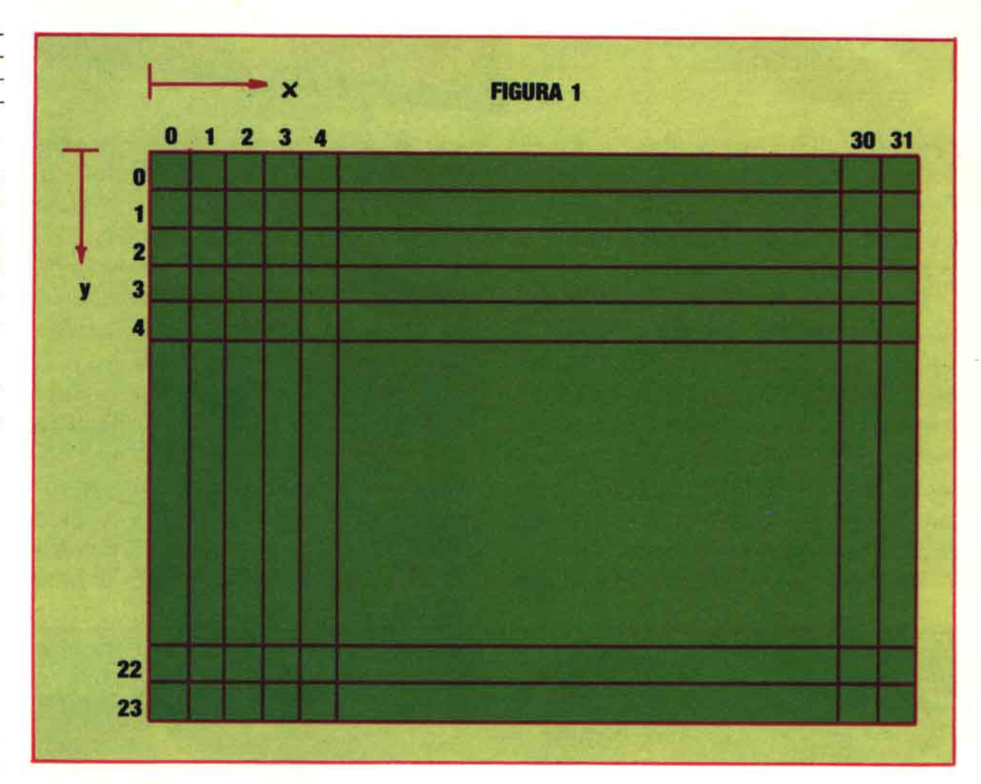
Con lo que comprobamos que el bloque dado por BASE(12), que contiene la TABLA DEL GENERADOR DE PATRONES DEL MODO DE GRAFICOS DE ALTA RESOLUCION tiene una longitud de 6144 bytes (de 0 a 6143).

Para escribir algo en un byte de este bloque conviene hacerlo en base 2 y utilizar la función VPOKE de BASIC. Si, por ejemplo, queremos dibujar en el cuadro (10,10) una trama como la de la figura 4, podemos hacerlo con el siguiente programa:

```
10 SCREEN 2
20 FOR I=2640 TO 2647 STEP 2
30 VPOKE I,85
40 VPOKE I+1,170
50 VPOKE 8192+I,&HF4
70 NEXT I
200 GOTO 200
```

COLOR PARA LOS PUNTOS

Hasta ahora hemos visto cómo dibujar puntos, pero nada hemos dicho del color. Al situar los puntos de la trama a que se refiere el ejemplo anterior, todos ellos se han dibujado con el color de la tinta inicialmente asignado (al conectar el ordenador, el color de la tinta es 15=blanco, y e del fondo es 4=azul). De esta forma, donde hemos puesto un 1 se ha dibujado un punto de color blanco y donde hemos puesto un 0 se ha dibujado un punto de color azul. Pero ya sabemos que el BASIC **MSX** nos permite dibujar puntos y líneas de distintos colores. ¿Cómo se almacena esto en la VRAM?



Recordemos que la pantalla está dividida en 6144 rebanadas de 8 *pixels*, y que a cada una de esas rebanadas le corresponde un byte en la VRAM. Pues bien, existe otro bloque de la VRAM (cuya dirección de comienzo es BASE(11)=8192) que también tiene una longitud de 6144 bytes y que está organizada de igual forma. Es la TABLA DE COLORES DEL MODO DE GRAFICOS DE ALTA RESOLUCION. Cada byte de esta tabla puede considerarse dividido en dos partes iguales de 4 bits, que se conocen como *nibble* alto y *nibble* bajo (fig. 5) (Se llama *nibble* a la mitad de un byte).

Observa que un número binario de 4 dígitos puede tener un valor decimal entre 0 (0000) y 15 (1111) (en hexadecimal: 0 y F). Casualmente, los colores de que dispone tu **MSX** varían entre 0 (transparente) y 15 (blanco). Por tanto, por cada byte de la tabla 11 podemos almacenar dos colores, correspondientes al color del fondo (bits a 0) y al color de la tinta (bits a 1) de la tabla 12.

El octeto correspondiente al color de la rebanada k del cuadro (i,j), se obtiene sin más que sumar a la fórmula dada anteriormente la cantidad 8192:

```
BYTE COLOR=8192+256*j+8*i+k
```

Así, si en el ejemplo anterior queremos que las rebanadas pares (0, 2, 4 y 6) tengan un color de tinta blanco (15=&HF) y un color de fondo negro (1=&H1), y las rebanadas impares (1, 3, 5 y 7) tengan un color de tinta rojo (6=&H6) y un color de fondo amarillo (10=&HA), basta modificar el programa añadiendo las dos nuevas líneas siguientes:

```
50 VPOKE 8192+I,&HF1
60 VPOKE 8193+I,&H6A
```

Cuando se trata de color, es conveniente escribir el valor del byte en forma hexadecimal. El primer dígito indicará así el color de la tinta y el segundo el color del fondo (p.e.: &HF1 = blanco, negro).

REPITIENDO CUADRO

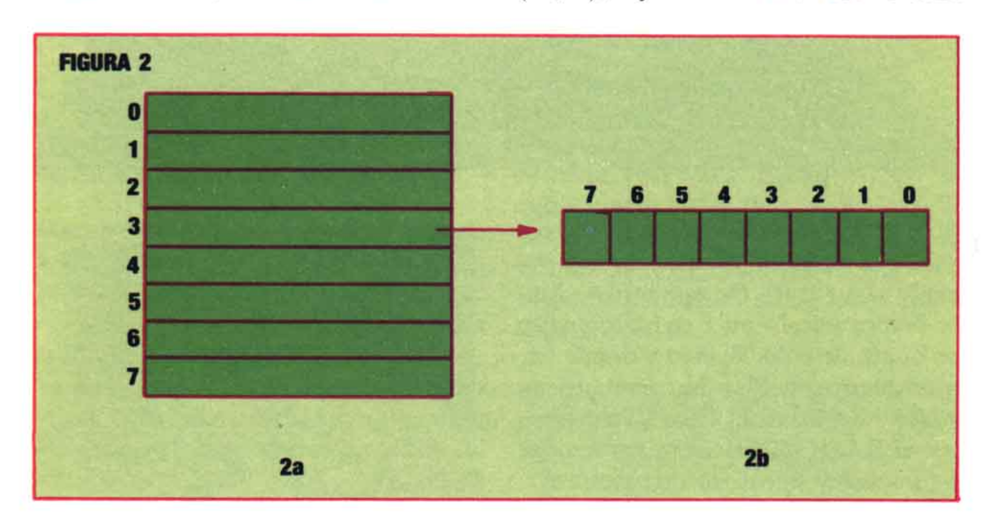
La pantalla de alta resolución está dividida en tres franjas horizontales.

En cada franja horizontal podemos repetir cuadros de la cuadrícula que habíamos definido al principio sin necesidad de definirlos más que una sola vez, como veremos ahora.

Cada franja horizontal tiene 32 cuadros en sentido horizontal (256 *pixels*) y 8 cuadros en sentido vertical (64 *pixels*) de forma que cada una de ellas contiene 256 cuadros. La primera franja es la zona correspondiente al tercio superior de la pantalla, la segunda al tercio intermedio y la tercera al tercio inferior.

La tabla 10, TABLA DE NOMBRES DE PATRONES DEL MODO DE GRAFICOS DE ALTA RESOLUCION, contiene las direcciones correspondientes a las tres zonas. La primera zona empieza en la dirección 6144 (BASE (10)), la segunda en la 6400 y la tercera en la 6656.

La ordenación de cada una de estas zonas es parecida a la que hemos visto al hablar de la tabla 12. En la zona 1, al grupo de 8 rebanadas pertenecientes al cuadro (0,0) le corresponde el byte 6144+0 de la tabla 10, al cuadro (1,0), el 6144+1, ... al cuadro (31,0), el 6144+31, al cuadro (0,1), el byte 6144+32, y así sucesivamente hasta llenar la primera zona (cuadro (31,7), byte 6144+255). La zona 2 comienza en el cuadro (0,8), al que corresponde el byte 6144+256 (o bien 6400+0), y termina en el cuadro (31,15), byte 6144+256+255 o bien (6400+255). La zona 3 comienza en el cuadro (0,16), byte 6144+512 (o bien 6656+0) y termina en el cuadro (23,31), byte 6144+512+255 (o bien 6656+255).



Normalmente, en cada byte de cada una de las tres zonas habrá un número, comprendido entre 0 y 255, igual al valor del lugar que ocupa el byte dentro de la zona. Así, en el octeto 6144 habrá un 0, en el 6145, un 1, ... en el 6399, un 255, en el 6400, un 0, ... en el 6655, un 255, en el 6656, un 0,... y en el 6911, un 255.

El ordenador interpreta estos valores de la siguiente forma: si en el byte 6144 hay un 0, se dibujan, en el cuadro (0,0) de la pantalla los bytes 0 al 7 (Tabla 12), con los colores indicados en los bytes 8192 al 8199; pero si hay un 1, por ejemplo, en el cuadro (0,0) se dibujarán los bytes 8 al 15 con los colores indicados en los bytes 8200 al 8207.

Si en el byte 6400 (primero de la zona 2) hay un 0, se dibujará lo que hay entre los bytes 2048 y 2055, con los colores indicados en los bytes 10240 al 10247, etcétera.

Al cuadro (i,j) (según lo definimos anteriormente), le corresponde, en la tabla 10, el byte:

```
BYTE(10)=6144+32*j+i
```

y la posición que le corresponde dentro de su zona es:

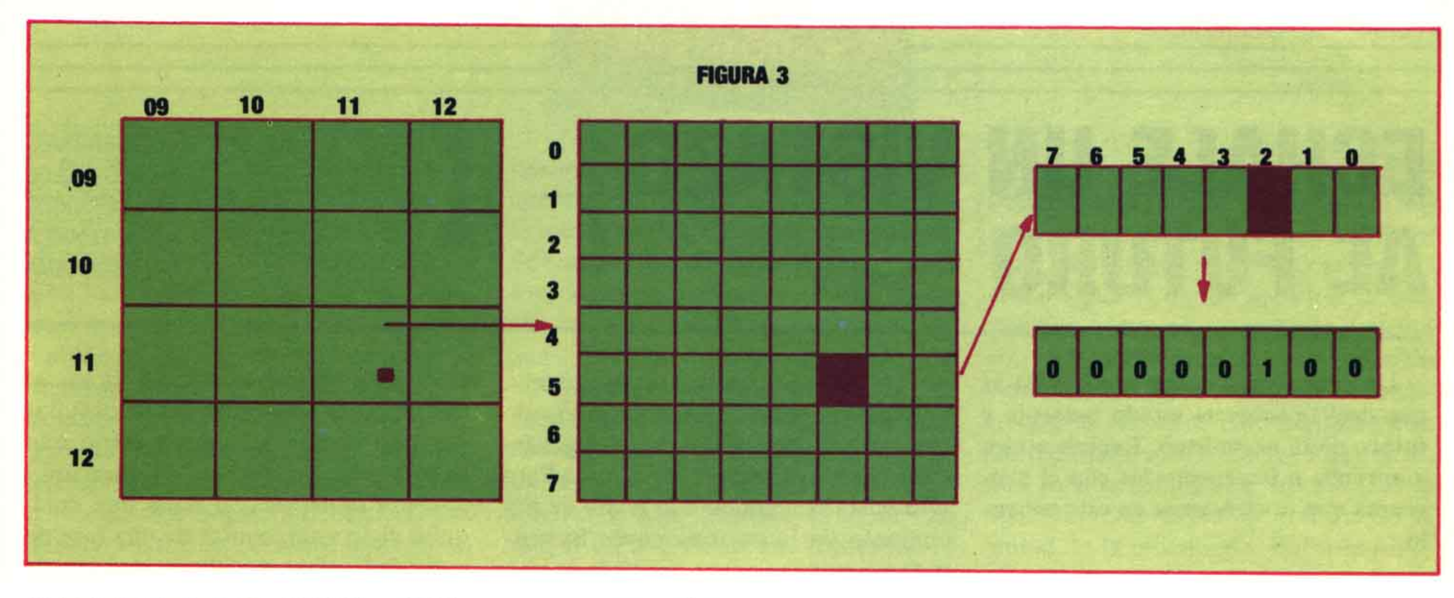
```
POSICION=BYTE(10)-
DIRECCION INICIO ZONA
```

Así, si queremos dibujar el cuadro (i,j) con los puntos y colores correspondientes al cuadro (m,n), tendremos que poner en el byte correspondiente a (i,j) el valor de la posición del cuadro (m,n), es decir:

GANADORES DE LOS MEJORES DE INPUT MSX

En el sorteo correspondiente al número 2 entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
Ivan Valdemoros Manchado	Bilbao	Hypersport 1
Antonio Aguilar Crespi	Barcelona	Hypersport 1
Federico González Sánchez	La Bisbal (Gerona)	Master of the Lamps
J. Miguel Rodríguez López Cano	Consuegra (Toledo)	Profanation
David Lindemann Rogner	Madrid	Road Fighter
J. Vicente Planells Alagarda	La Punta (Valencia)	Golf
Ricardo Sotomayor Alfonso	Sevilla	Zakil Wood
Christian José Montejo	La Coruña	Gunfright
Luis Hidalgo García	Madrid	Boxing
Xavier Sánchez Perchachs	S. M.ª de Palautordera (Barcelona)	Hyper Rally




```
6144+32*N+M-DIR.INICIO ZONA
```

Evidentemente eso sólo es posible cuando los cuadros (i,j) y (m,n) están en la misma zona.

Así, por ejemplo, el cuadro (10,10) de ejemplo que hemos venido manejando, está en la zona 2, y su posición es:

```
POSICION(10,10)=6144+
32*10+10-6400=74
```

Si queremos que toda la zona 2 aparezca con la trama de dicho cuadro, basta añadir a nuestro programa las líneas:

```
80 FOR I=0 TO 255
90 VPOKE 6400+I,74
100 NEXT I
```

UNA APLICACION: COMO ALMACENAR LA PANTALLA EN RAM

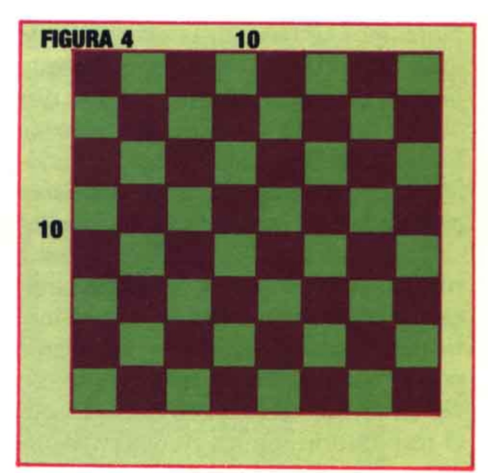
A continuación listamos una rutina en BASIC que nos permitirá almacenar los bloques correspondientes a las tablas 10, 11 y 12 en RAM, es decir, guardar la pantalla gráfica. Se puede salvar en cinta magnética utilizando la instrucción:

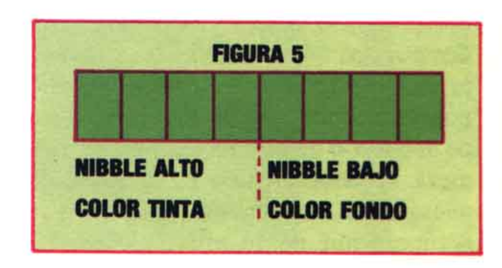
```
BSAVE"CAS:VRAM",48900,61968
```

Teclea lo que sigue:

```
10 CLEAR 200,48899!
50000 'VRAM A RAM
50010 FOR I=0 TO 6143
50020 POKE 48900!+I,
      VPEEK(I)
50030 POKE 55050!+I,
      VPEEK(8192+I)
50040 NEXT I
50050 FOR I=0 TO 767
50060 POKE 61200!+I,
      VPEEK(6144+I)
50070 NEXT I
50080 RETURN
60000 'RAM A VRAM
60010 FOR I=0 TO 6143
60020 VPOKE 8192+I,
      PEEK(55050!+I)
60030 VPOKE I,PEEK
      (48900!+I)
60040 NEXT I
60050 FOR I=0 TO 767
60060 VPOKE 6144+I,
      PEEK(61200!+I)
60070 NEXT I
60080 RETURN
```

Estas rutinas tardan, cada una, 1 minuto y 40 segundos en ejecutarse. Para utilizarlas tendrás que llamarlas con GO SUB, una vez que tengas a la vista el dibujo que quieras guardar. El siguiente programa en BASIC contiene rutinas en código máquina que se ocupan de realizar las mismas operaciones en ¡UN QUINTO! de segundo.






```
10 CLEAR 200,48899!
20 FOR I=1 TO 74
30 READ A$
40 POKE 62000!+I,VAL
   ("&H"+A$)
50 NEXT I
60 DEFUSR=62001!
   :' VRAM A RAM
70 DEFUSR1=62038!
   :' RAM A VRAM
10000 DATA 21,00,00,01,00,18,
      11,04,BF,CD,59,00,21,00
      ,20,01,00,18,11,0A,D7,C
      D,59,00,21,00,18,01,00,
      03,11,10,EF,CD,59,00,C9
10010 DATA 21,04,BF,01,00,18,
      11,00,00,CD,5C,00,21,0A
      ,D7,01,00,18,11,00,20,C
      D,5C,00,21,10,EF,01,00,
      03,11,00,18,CD,5C,00,C9
```

Para ejecutarlas basta hacer U=USR(0), para pasar la VRAM a RAM; y U=USR1(0) para pasar la RAM a VRAM.

ECHALE UN VISTAZO AL FUTURO

Los biorritmos tienen muchas cosas que decirte sobre el estado presente y futuro de tu organismo. Represéntalos y aprende a interpretarlos con el programa que te ofrecemos en este artículo.

Toda la interesante teoría de los biorritmos se basa en la existencia de unas energías, o más propiamente bioenergías, que surgen del interior del ser humano y que evolucionan de forma periódica. Esta evolución periódica determina un ciclo de dos fases para todas las bioenergías, ciclo relacionado con los innumerables ciclos o ritmos vitales a que se ve sujeto cualquier ser vivo y que suelen ser producto de procesos orgánicos. Pensemos por un momento en el ciclo o ritmo de vigilia/sueño, en el ritmo cardiaco, en el respiratorio, en los ritmos determinados por las ondas cerebrales... Todos ellos son ritmos vitales constituidos por dos fases distintas que, en una primera aproximación, podemos denominar como: fase de actividad y fase de reposo o recuperación. En la primera lo característico es el gasto de energía, en la segunda predomina la recuperación de la energía gastada. Hay muchos tipos de ciclos vitales. La mayoría son ciclos con una duración igual o inferior a un día, por ejemplo el ciclo vigilia/sueño. También hay ciclos de mayor duración que, en general, son menos conocidos y más difíciles de estudiar. Es en este tipo de ciclos en donde la teoría de los biorritmos incluye a sus tres ciclos vitales, a los que denomina biorritmos físico, emocional e intelectual. La teoría afirma que estos tres biorritmos representan el gasto de bioenergía, en cada uno de los tres aspectos citados, por parte de todo ser humano. Como los tres biorritmos son periódicos (se repiten al transcurrir un cierto número de días) y como además dan comienzo el día de nacimiento de la persona, es bastante sencillo calcular el estado de cada uno de ellos para cualquier fecha; pasada, presente o ¡futura! Para ello basta con conocer la fecha de nacimiento de la persona cuyos biorritmos quieres examinar. El resto es una pura y simple labor de cálculo, sumamente tediosa si tienes que hacerla a mano, pero muy sencilla si utilizas tu **MSX** y el programa que incluimos en este artículo. Con él y con los postulados de la teoría de los biorritmos, te vas a poder convertir en un explorador del futuro, conociendo de antemano el estado que tendrán tus biorritmos (o los de cualquier otra persona cuya fecha de nacimiento conozcas) pasado mañana, el mes que viene o dentro de algunos años. Con esta información podrás distribuir tus esfuerzos, estableciendo tiempos de actividad y reposo que estén de acuerdo con tus posibilidades bioenergéticas de cada momento.

LOS TRES CICLOS

La teoría de los biorritmos hace hincapié en los ritmos que miden los niveles de bioenergía de las actividades física, emocional e intelectual. Estos biorritmos son fácilmente representables poque tienen periodos de un número exacto de días. El físico es el más corto, con una duración de 23 días. Le sigue el emocional, de 28 días y por último el intelectual, de 33 días. Los tres dan comienzo en la hora cero del día de nacimiento de la persona, y comienzan su evolución por la fase positiva o de gasto bioenergético. Al tener periodos distintos, las curvas que los representan van evolucionando de forma diferente, cruzándose y adoptando todas las combinaciones de posición relativa posibles. Con el paso del tiempo legan a repetirse las posiciones adoptadas por cada ciclo, lo que da lugar a otros «superciclos»; por ejemplo el que se conoce como año biorrítmico, de 664 días de duración.

Cada biorritmo, al igual que cualquier ciclo vital, consta de una fase de actividad o fase positiva y de otra de reposo o fase negativa. Si nos encontramos en fase negativa, estamos en malas condiciones para exigirle a nuestro organismo, convendrá que reduzcamos nuestra actividad para poder recuperarnos. Por el contrario, en la fase positiva o de gasto de bioenergía, estaremos en buenas condiciones para desarrollar al máximo nuestra actividad.

Vamos a ser más concretos examinando uno a uno los tres ciclos.

El ciclo físico se considera relacionado con la actividad muscular. Es el que representa la capacidad para el esfuerzo físico, por ello es especialmente importante para las personas que realizan trabajos o ejercicios físicos de una forma continuada.

El ciclo emocional, al que también se conoce como ciclo femenino, está relacionado con el sistema nervioso y se puede considerar responsable de la actitud emocional en aspectos como la capacidad creadora, el entusiasmo, la capacidad de relación personal, etc. Es el ciclo más importante en cuanto consideramos actividades artísticas y creadoras o, en general, cualquier actividad en la que prime el factor emocional.

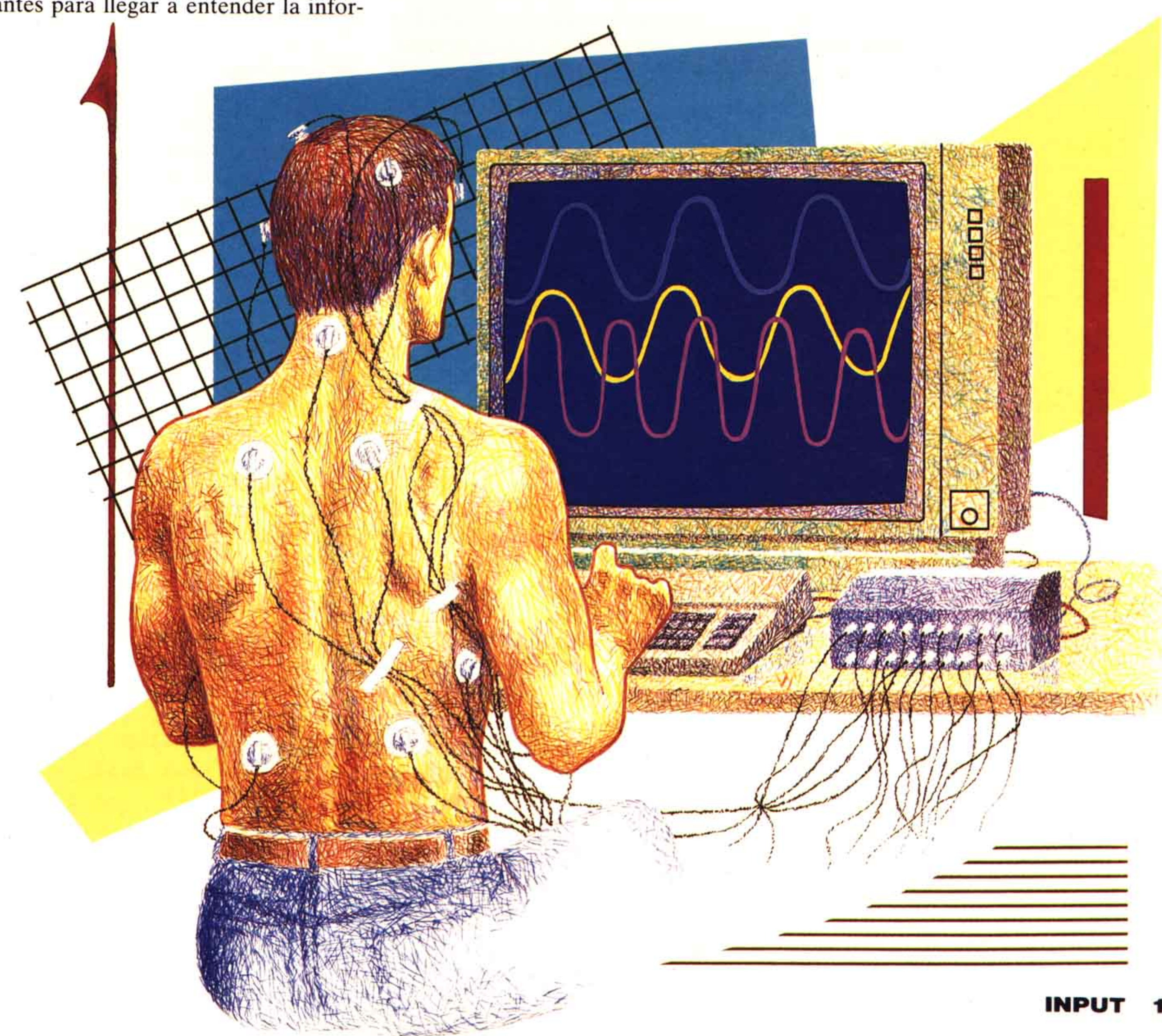
En cuanto al ciclo intelectual, el más largo de todos, se le considera relacionado con la actividad mental, con la capacidad para pensar, razonar, analizar, recordar, asimilar... es decir, con todas aquellas actividades en las que se utiliza el cerebro de una forma consciente. Es el ciclo más importante si deseamos conocer nuestro estado en todo lo relacionado con las actividades intelectuales.

Hemos incluido un cuadro resumen con las caracteristicas de cada uno de los ciclos. En cualquier caso hay que tener en cuenta que los biorritmos dan una indicación de el estado de actividad o reposo bioenergético, es decir hablan en un sentido muy general, se refieren a tendencias, propensiones o actitudes más que a hechos concretos. Por ello no tiene ningún sentido, por ejemplo, relacionar un dolor de cabeza con la fase negativa del ciclo físico. Hablaremos más adelante sobre el aspecto de la interpretación de los biorritmos.

REPRESENTACION GRAFICA

Uno de los apartados más importantes para llegar a entender la información de los biorritmos es tener en cuenta cual es su representación gráfica (a la que se denomina biograma). Se han propuesto varias fórmulas para los biogramas: representaciones lineales en un calendario, líneas en zigzag... Pero sin duda, la representación más popular es la sinusoidal, en la que cada biorritmo se representa por una sinusoide (una función seno) que se inicia, con fase cero, el día de nacimiento y cuyo periodo es el mismo que el del biorritmo que representa.

Aunque es la más popular, hay que tener cuidado con ella, pues es fácil que lleve a malas interpretaciones. Intuitivamente se tiende a pensar que cuando la función seno llega a su máximo es porque nuetra bioenergía es también máxima y lo mismo; que el mínimo de la función representa el mínimo bioenergético. En realidad no es así. La interpretación correcta es como sigue: Cuando se inicia la fase positiva, esto es, cuando la función seno cruza el eje para iniciar su ciclo positivo, es cuando el nivel de bioenergía es máximo. Esta bioenergía se va consumiendo a lo largo del ciclo, con lo que al llegar al máximo de la función seno ya se ha consumido la mitad de la bioenergía. Por lo mismo, el mínimo de bioenergía no se da en el mínimo de la función, sino al inicio de la fase negativa o de recuperación, que es cuando la función cruza el eje para iniciar su ciclo negativo.



RITMO	INFLUENCIA	FASE POSITIVA	DIA CRITICO	FASE NEGATIVA
Fisi.	Fuerza Aguante Vigor Confianza personal	Plenitud de fuerzas Competiciones Viajes Operaciones quirurgicas	Estado inestable Accidentes Baja forma fisica	Cansancio Falta de vigor
Emo.	Sentimientos Emociones Humor Intuicion	Trabajo en equipo Relaciones personales Citas	Emociones inestables Disputas Humor variable	Monotonia Relaciones personales flojas
Int.	Logica Ingenio Memoria Concentracion	Estudio Toma de decisiones Hacer planes Examinarse	Falta de atencion Errores Poca memoria	Escasa capacidad racional Poner orden en las ideas

Cuando se llega al mínimo de la función, ya se ha recuperado la mitad de la bioenergía.

Teniendo en cuenta este pequeño detalle de interpretación, se puede sacar el máximo partido de la representación por funciones seno que, por otra parte, es la más atractiva estéticamente y la que mejor refleja la diferencia entre el ciclo positivo (por encima del eje) y el negativo (por debajo). Esta forma de representar los biorritmos es la que hemos escogido y la que utiliza el programa.

LOS DIAS CRITICOS

Los días críticos son aquellos en los que se produce la transición entre las fases positiva y negativa de un biorritmo. En la representación mediante funciones seno, esto corresponde a los días en los que la función pasa por cero (cuando cruza el eje). En cada ciclo completo de cada biorritmo hay dos días críticos diferentes. El primero es el día crítico positivo, en la transición de la fase negativa a la positiva. El segundo, que se denomina día crítico negativo, corresponde a la transición inversa, es decir, de la fase positiva a la negativa. Según la teoría, los días críticos son días caracterizados por la inestabilidad. El organismo está en un momento de transición y tiene que adaptarse a las nuevas condiciones. En el día crítico positivo, el organismo está al máximo de bioenergía, pero pasa de la fase de recuperación a la de gasto de la misma. En el día crítico negativo la situación es todavía peor, pues la transición de la fase de gasto a la de recuperación se produce cuando el organismo está al mínimo de bioenergía. Esto hace que el día crítico negativo se considere peor que el positivo; al tener menos bioenergía, el organismo tiene menos capacidad de reacción y tarda más en adaptarse al cambio que se está produciendo.

Los investigadores del tema, con las estadísticas en la mano, consideran que en los días críticos hay mayor propensión a que las cosas vayan mal. En el ciclo físico y en estos días, hay mayor propensión a los accidentes o a contraer enfermedades. En el ciclo emocional hay más probabilidades de tomar decisiones precipitadas y poco razonadas. En el intelectual es más probable que fallen la capacidad de razonar, la memoria, la concentración etc.

Como verás los augurios no son muy buenos para los días críticos. Pero todavía peor que un día crítico, dicen los expertos, es un día crítico doble o triple, esto es, un día en el que coincidan los días críticos de dos o más de los biorritmos. En este caso la inestabilidad será doble o triple y cualquiera sabe lo que puede pasar.

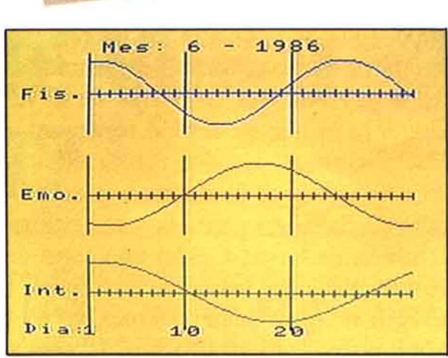
De todas formas, conviene tomar las cosas con serenidad y no asustarse. Si observamos un poco, comprobaremos que el número de días críticos es de unos seis al mes, algo más de uno por semana. Y hay semanas, o meses, en los que no nos ocurre nada que pueda considerarse como «malo». Con esto queremos decir que según la teoría de los biorritmos un día crítico no significa que vayamos a tener un accidente. Lo único que ocurre en un día crítico es que nos encontraremos un tanto confusos y nuestra capacidad se verá ligeramente disminuida.

EL PROGRAMA

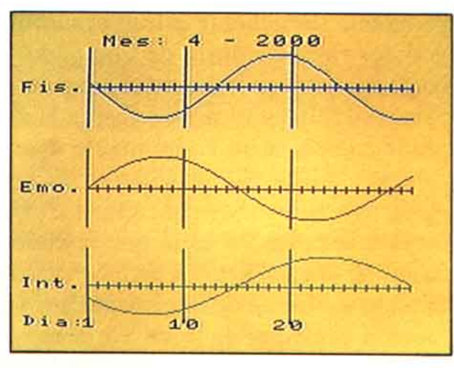
Después de tanta teoría vamos a pasar a la práctica. Para ello lo primero que tienes que hacer es teclear el siguiente programa.

```
10 GOSUB 4000
20 GOSUB 2000
30 GOSUB 1000
40 GOSUB 3000
50 IF T$="s" THEN RUN
   ELSE END
1000 'calculo dias
     transcurridos
1005 '
1010 DT=0
1011 IF AA=AN THEN
     GOTO 1300
1012 IF AA=AN+1 THEN
     GOTO 1050
1015 '
1020 'anos completos
1025 FOR J=AN+1 TO AA-1
1030 IF (JMOD4)=0 THEN
     DT=DT+366 ELSE
     DT=DT+365
1035 NEXT J
1040 '
1045 'meses completos
1046 IF MN=12 THEN 1066
1050 FOR J=MN+1 TO 12
1055 DT=DT+ME(J)
```

```
1060 NEXT J
1065 IF (ANMOD4)=0 AND
     MN+1<=2 THEN DT=DT+1
1066 IF MA=1 THEN GOTO
     1100
1070 FOR J=1 TO MA-1
1075 DT=DT+ME(J)
1080 NEXT J
1085 IF (AAMOD4)=0 AND
     MA-1>=2 THEN DT=DT+1
1090 '
1095 'dias completos
1100 DT=DT+(ME(MN)-DN+1)
1105 IF (ANMOD4)=0 AND
     MN=2 THEN DT=DT+1
1110 DT=DT+DA-1
1115 RETURN
1300 IF MA=MN THEN GOTO
     1400
1305 IF MA=MN+1 THEN
     GOTO 1330
1310 FOR J=MN+1 TO MA-1
1315 DT=DT+ME(J)
1320 NEXT
1325 IF (ANMOD4)=0 AND
     MN=1 THEN DT=DT+1
1330 DT=DT+(ME(MN)-DN+1)
1335 IF (ANMOD4)=0 AND
     MN=2 THEN DT=DT+1
1340 DT=DT+DA-1
1345 RETURN
1400 IF DA=DN THEN RETURN
1405 DT=DA-DN
1410 RETURN
2000 'lectura de fechas
2005 '
2010 CLS:LOCATE 3,2:PRINT
     "ESCRIBE LA FECHA DE
     NACIMIENTO":LOCATE 3,3:
     PRINT"======= == =====
     == =========="
2015 P=5:GOSUB 2105
2020 AN=A:MN=M:DN=D
2025 LOCATE 3,9:PRINT"ESCRIBE
     LA FECHA ACTUAL":LOCATE
     3,10:PRINT"======= == ==
     === ======"
2030 P=12:GOSUB 2105
2040 AA=A:MA=M:DA=D
2045 IF AA<AN THEN GOTO
     2010
2050 IF AA=AN AND MA<MN THEN
     GOTO 2010
2055 IF AA=AN AND MA=MN
     AND DA<DN THEN GOTO
     2010
```




```
2060 RETURN
2100 LOCATE 0,P:PRINT
     SPC(30)
2105 LOCATE 0,P:INPUT
     "ANO (AAAA) ";A$:IF
     LEN(A$)<>4 THEN 2100
2110 A=VAL(A$)
2115 LOCATE 0,P+1:PRINT
     SPC(30)
2120 LOCATE 0,P+1:INPUT"MES
     (MM) ";M$:IF LEN(M$)
     <>2 THEN 2115
2125 M=VAL(M$):IF M<1 OR
     M>12 THEN 2115
2130 LOCATE 0,P+2:PRINT
     SPC(30)
2135 LOCATE 0,P+2:INPUT"DIA
     (DD) ";D$:IF LEN(D$)
     <>2 THEN 2130
2140 D=VAL(D$):IF D<1 OR
     D>31 THEN 2130
2145 RETURN
3000 'presentacion en
     SCREEN2
3005 '
3010 DR=DT-DA+1
3015 SCREEN 2:COLOR 1,15,15
     :CLS:OPEN "grp:" AS#1
     :PI=3.14159:H=20
3020 'ejes
3025 CO=4:XI=50:XF=200:RE=5
3030 LINE (XI,32)-(XF,32),CO
3035 FOR E=XI TO XF STEP RE
     :LINE(E,30)-(E,34),CO
     :NEXT
3040 LINE (XI,7)-(XI,57),CO
3042 LINE (95,7)-(95,57),CO:
     LINE(145,7)-(145,57),CO
3045 CO=6
3050 LINE (XI,96)-(XF,96),CO
3055 FOR E=XI TO XF STEP RE:
     LINE(E,94)-(E,98),CO
     :NEXT
3060 LINE(XI,71)-(XI,121),CO
3062 LINE(95,71)-(95,121),CO:
     LINE (145,71)-(145,121),
     CO
3065 CO=12
3070 LINE  (XI,160)-(XF,160),
     CO
3075 FOR E=XI TO XF STEP RE
     :LINE(E,158)-(E,162),CO:
     NEXT
3080 LINE  (XI,135)-(XI,185),
     CO
```




```
3082 LINE(95,135)-(95,185),CO
     :LINE(145,135)-(145,185)
     ,CO
3085 'textos
3090 DRAW"BM57,0":PRINT#1,
     "Mes:";MA;"-";AA
3095 DRAW"BM18,28":PRINT#1,
     "Fis."
3100 DRAW"BM18,92":PRINT#1,
     "Emo."
3105 DRAW"BM18,156":PRINT#1,
     "Int."
3110 DRAW"BM19,183":PRINT#1,
     "Dia:"
3112 DRAW"BM48,183":PRINT#1,
     "1"
3115 DRAW"BM89,183":PRINT#1,
     "10"
3120 DRAW"BM138,183":PRINT#1,
     "20"
3240 'dibujo biorritmos
3245 FOR J=50 TO 200
3250 X=DR+(J-50)/5
3255 Y1=32-H*SIN(X/23*2*PI)
3260 Y2=96-H*SIN(X/28*2*PI)
3265 Y3=160-H*SIN(X/33*2*PI)
3270 PSET(J,Y1),4:PSET(J,Y2),
     6:PSET(J,Y3),12
3275 NEXT
3280 DRAW"BM97,117":PRINT#1,
     "OTRO(S/N)"
3285 T$=INKEY$:IF T$="" THEN
```

```
3285 ELSE RETURN
4000 'inicializacion
4005 '
4010 SCREEN0:CLS:KEY OFF
     :DIM ME(12)
4015 FOR J=1 TO 12
     :READ ME(J):NEXT
4020 DATA 31,28,31,30,31,30,3
     1,31,30,31,30,31
4025 RETURN
```

Comprobarás que se trata de un programa sumamente estructurado. A grandes rasgos, consta de una pequeña parte, correspondiente al programa principal (líneas 10 a 50) encargada de pasar control a un conjunto de 4 subrutinas. Cada una de ellas se encarga de una labor muy concreta. La primera que se utiliza es la que comienza en la línea 4000. La hemos denominado subrutina de inicialización y su función principal es la de almacenar, en la matriz ME, el número de días de cada uno de los meses del año. Después de ella, el programa transfiere control a la que empieza en la línea 2000. Esta es la encargada de recoger los datos sobre la fecha de nacimiento de la persona, así como los de la fecha para la que se desean representar los biorritmos. Es una rutina dedicada sobre todo al formateo y presentación de datos en pantalla y se encarga también de llevar a cabo una serie de comprobaciones sobre las fechas, para asegurar que sean correctas. Por ejemplo no admitirá fechas de representación anteriores a la fecha de nacimiento, ni valores para el día del més que no sean correctos (por ejemplo un 34 de enero). Tras esta entrada de datos, el programa da paso a la subrutina que comienza en la línea 1000. Esta es la más importante de todas ya que se encarga de calcular el número de días que han transcurrido (DT) desde la fecha de nacimiento a la fecha de representación. Es una subrutina compleja, ya que tiene que tener en cuenta factores como los años bisiestos, a la hora de llevar a cabo los cálculos.

Por último, el programa da paso a la subrutina de presentación que es la encargada de dibujar las curvas de los tres biorritmos. Para ello se hace uso del modo gráfico de alta resolución SCREEN 2.

¿VERDAD O MENTIRA?

La validez de la teoría de los biorritmos no es en modo alguno incuestionable. Son muchas y muy serias las objeciones a todo el entramado de la teoría. Hay mucha gente que cree en el valor científico de los biorritmos, pero también hay mucha que niega esta validez. Quizá como dato pueda servirnos el de que algunas compañías aéreas de prestigio internacional utilcen los biorritmos para programar las fechas de vuelo de sus pilotos.

En **INPUT** no nos pronunciamos ni en un sentido ni en otro. Unicamente nos ha parecido un tema interesante y, además, un excelente ejercicio de programación.

¿MONITOR DE VIDEO O TELEVISOR?

- COMO FUNCIONAN
- TIPOS DE ENTRADAS DE SEÑAL
- MONOCROMO Y COLOR
- DEFINICION DE LA IMAGEN

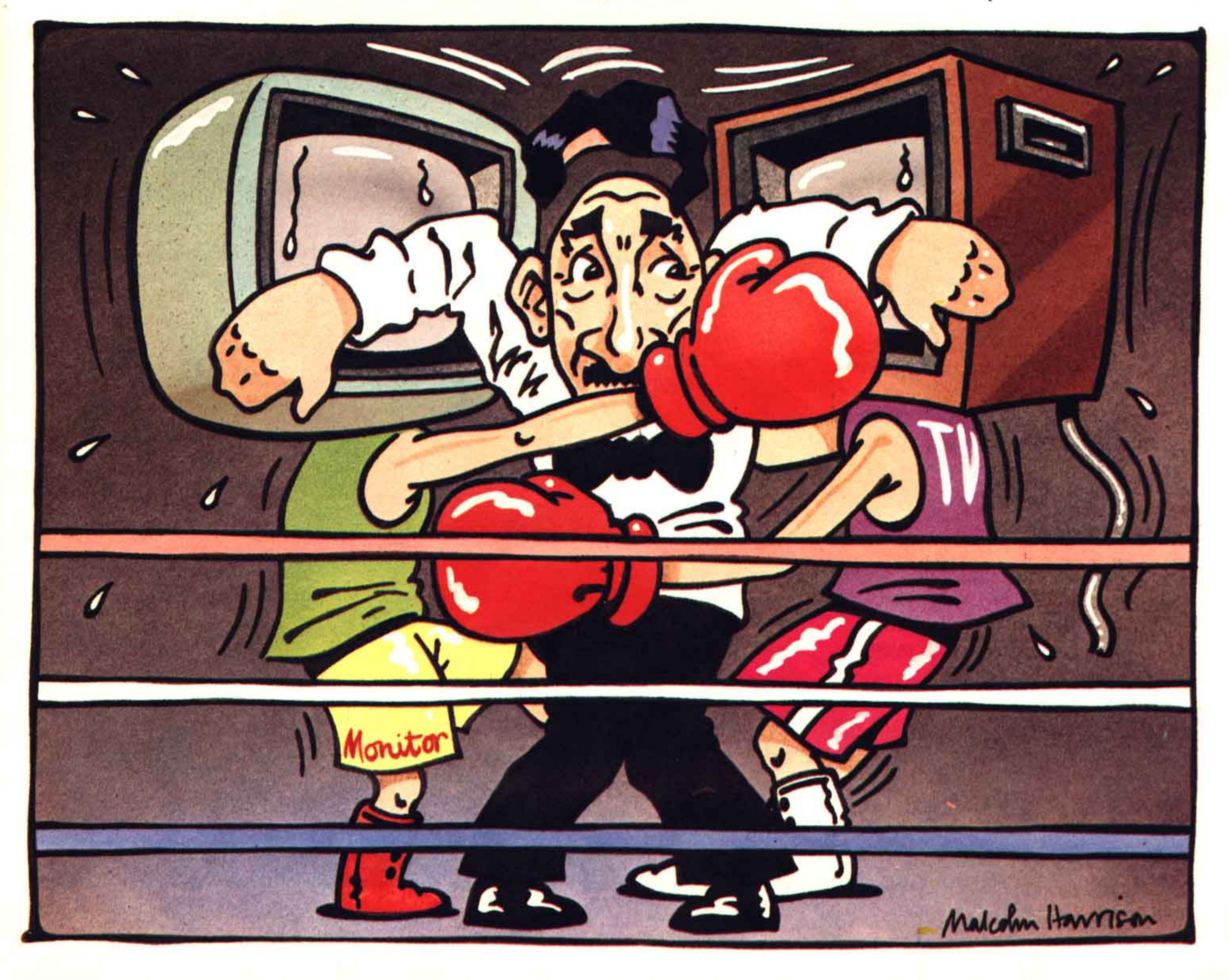
Puede ser difícil realizar una elección entre un televisor de calidad y un monitor de video, pero en muchos casos no cabe más que una opción. En este artículo dedicamos nuestra atención a algunas de las diferencias generales entre los dos tipos.

La pantalla es un elemento vital de cualquier ordenador, aunque muchas veces se la considera como una especie de mula de carga en vez del sofisticado elemento electrónico que en realidad es. La pantalla es el principal canal de comunicaciones entre el ordenador y nosotros. En el otro sentido la información fluye desde el usuario hacia el ordenador, normalmente a través del teclado, representando también la pantalla en este caso una réplica o eco de la información que se tecleó.

Si estás en una situación tan afortunada como para tener que escoger una nueva pantalla, o simplemente deseas aprovechar lo mejor posible la que tienes, te resultará conveniente, si no esencial, tener algunas nociones sobre la forma en que funciona una pantalla y la forma en que el ordenador controla la salida de video. En particular esto último es algo que afectará a tu elección.

PRINCIPIOS BASICOS

Aunque hay varias formas diferentes de transferir la información desde el ordenador a la pantalla, una cosa es común a todos los tipos de pantallas utilizados en aplicaciones tanto co-



merciales como industriales o domésticas: el tubo de rayos catódicos, también conocido por sus iniciales inglesas CRT. La base del funcionamiento de los tubos de rayos catódicos reside en el hecho de que ciertos compuestos de fósforo emiten luz cuando son excitados por un haz de electrones.

Si se dispara un chorro de electrones con un cañón electrónico sobre una lámina de vidrio cubierta con fósforo, cada vez que se produzca un disparo el fósforo brillará. Si se desplaza el cañón de forma que el chorro de electrones vaya recorriendo toda la pantalla, será toda la superficie de la misma lo que emitirá brillo al estar sometida al bombardeo de electrones. Si se hace que el haz de electrones se mueva, por ejemplo, describiendo la figura de un ocho, aparecerá en la pantalla la figura de un ocho.

Un tubo de rayos catódicos es un cono de vidrio cerrado en cuyo interior se ha hecho el vacío, provisto de un cañón electrónico en el extremo próximo al vértice y con su base cubierta de fósforo.

Imaginemos un trozo de cartulina en el que se ha recortado la forma de un hombre, interponiéndose entre el cañón de electrones y la pantalla. La pantalla sólo brillará en los puntos en que el chorro de electrones no haya sido bloqueado por la cartulina. En consecuencia, la zona de pantalla cuyo fósforo brilla tendrá una forma parecida a la de un hombre. Se podría conseguir el mismo efecto interrumpiendo el haz electrónico (apagando el cañón) en los instantes adecuados, es decir cuando pasa por los puntos deseados de la pantalla.

Esta es exactamente la forma en que funcionan los televisores y la mayoría de los monitores, a excepción de algunos de construcción especial. Un cañón electrónico barre todo el ancho de la pantalla 625 veces para crear 625 líneas. (En realidad barre primero 313 líneas y a continuación rellena los 312 huecos que había dejado entre dichas líneas.)

El sistema no tiene partes mecánicas móviles, el desplazamiento del haz de electrones se consigue por medio de campos magnéticos. Por eso está recomendado no dejar materiales magnéticos —*diskettes* o cintas— cerca de un televisor. Para formar las imágenes, se conmuta el cañón de electrones encendiéndolo y apagándolo. La información que contiene la señal recibida por el televisor o el monitor, precisamente sirve para decirle al cañón de electrones cuándo tiene que emitirlos o no.

En lugar de recorrer toda la pantalla en lo que se llama un barrido por rastreo, algunos monitores producen la imagen guiando el haz de electrones por la pantalla como si fuera un lápiz, siguiendo las líneas de la imagen. Aunque se consiguen así resultados de excelente calidad, se trata de un procedimiento mucho más lento para usos generales. Aún suponiendo que el haz de electrones se desplazara por la pantalla a la nada despreciable velocidad de 40000 kilómetross por hora, una imagen de cierta complejidad aparecería ante el ojo humano como altamente insatisfactoria.

PERSISTENCIA DE LA VISION

Así se llama una de las características del ojo humano que en otras circunstancias podría parecer un defecto, pero que en este caso permite la utilización del tubo de rayos catódicos como instrumento de comunicación visual.

El ojo humano tiene la propiedad de mantener cualquier imagen de las que se forman sobre su retina durante un tiempo de aproximadamente 1/25 de segundo. El cañón de electrones que hay en la parte trasera de los tubos de rayos catódicos de los televisores domésticos y de la mayoría de los monitores, barre la superficie de la pantalla 625 veces en 1/50 de segundo. Esto significa que se forma una nueva semi-imagen o «trama» sobre la pantalla cincuenta veces por segundo. De esta forma aparece sobre la pantalla una imagen nueva antes de que se haya desvanecido la antigua de nuestro ojo. Por eso podemos ver una imagen que se mueve con suavidad.

Si nuestro ojo fuera capaz de retener la imagen solamente durante una centésima de segundo, la imagen de la pantalla nos parecería que parpadeaba de forma insoportable.

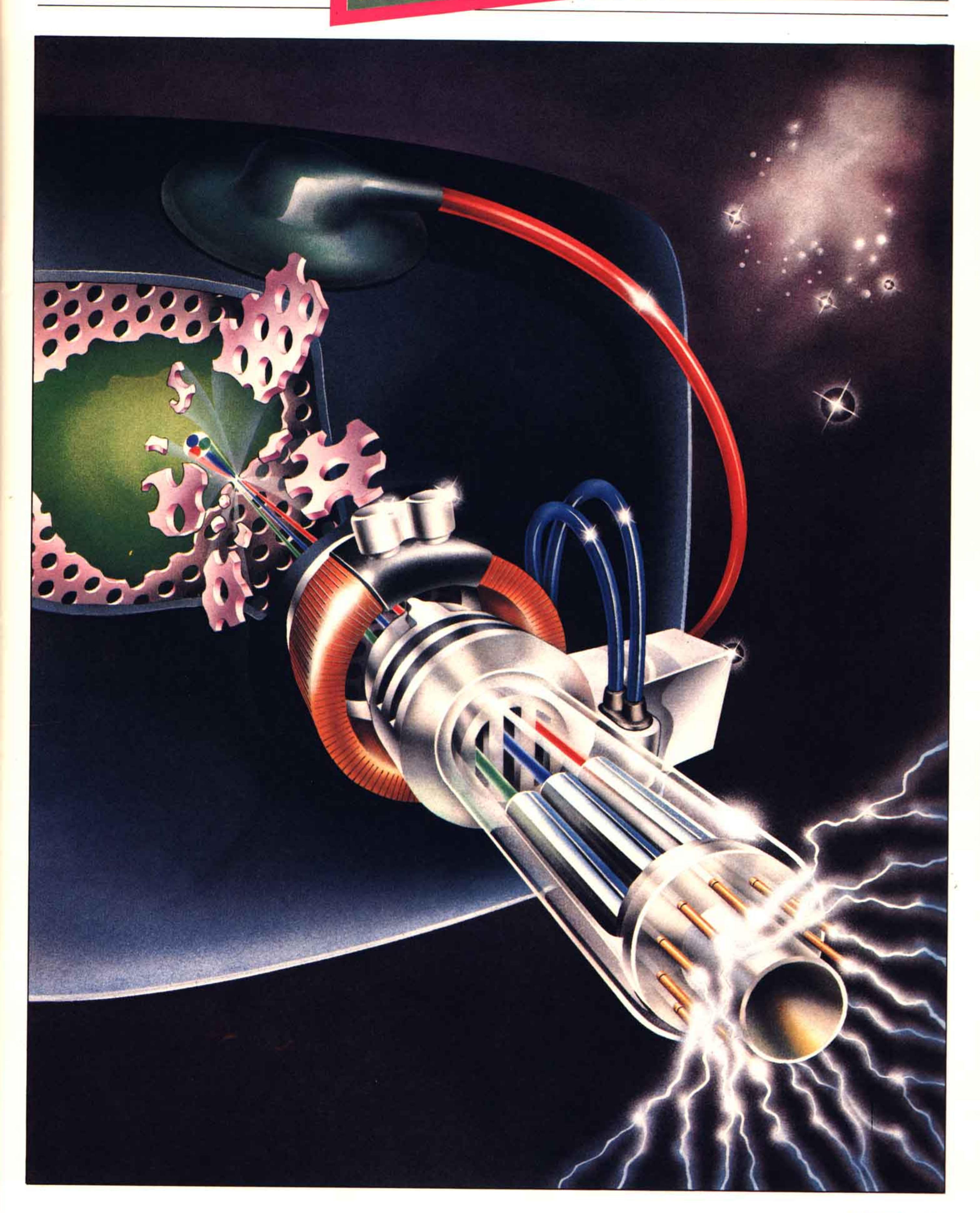
COLOR

El principio básico empleado es el mismo para televisores en blanco negro y color. La única diferencia entre ambos es que sólo se requiere un cañón de electrones y un tipo de recubrimiento de fósforo en el caso del monocromo, mientras que en el caso del color hacen falta tres tipos de fósforos diferentes en el recubrimiento de la pantalla y normalmente tres cañones electrónicos (hay televisores que tienen un solo cañón, pero que emiten tres haces de electrones).

El color se debe a que uno de los fósforos del recubrimiento produce luz roja, otro luz verde y el otro luz azul. Uno de los chorros de electrones activa el fósforo rojo, otro el fósforo verde y el tercero el fósforo azul. Combinando el rojo, verde y azul con diferentes intensidades se pueden obtener todos los demás colores e intensidades.

ENTRADA DE LA SEÑAL

Las pantallas, ya sean de un televisor o de un monitor, de color o de blanco y negro, pueden aceptar tres clases de señales de entrada. En primer lugar está la entrada normal de televisión enviada por las emisoras públicas, que es la señal que acepta el televisor de tu casa y produce en la pantalla las imágenes de televisión que estás acostumbrado a ver. Por otra parte está la llamada señal compuesta de videofrecuencia o vídeo compuesto. Esta es una de las señales que se emplea con más frecuencia en los monitores de vídeo y en la actualidad ya hay algunos televisores que tienen prevista una toma para aceptar dichas señales. La señal de vídeo compuesto contiene la información de la imagen y los impulsos de sincronismo. Esta última hace que el haz de electrones se sitúe en el lugar adecuado en cada momento.



Por último, está la llamada entrada RGB. Las letras corresponden a rojo, verde y azul (Red, Green, Blue). Es la forma más directa y más precisa de especificar la información de color. La información relativa a cada color se introduce directamente y por separado desde el ordenador al monitor.

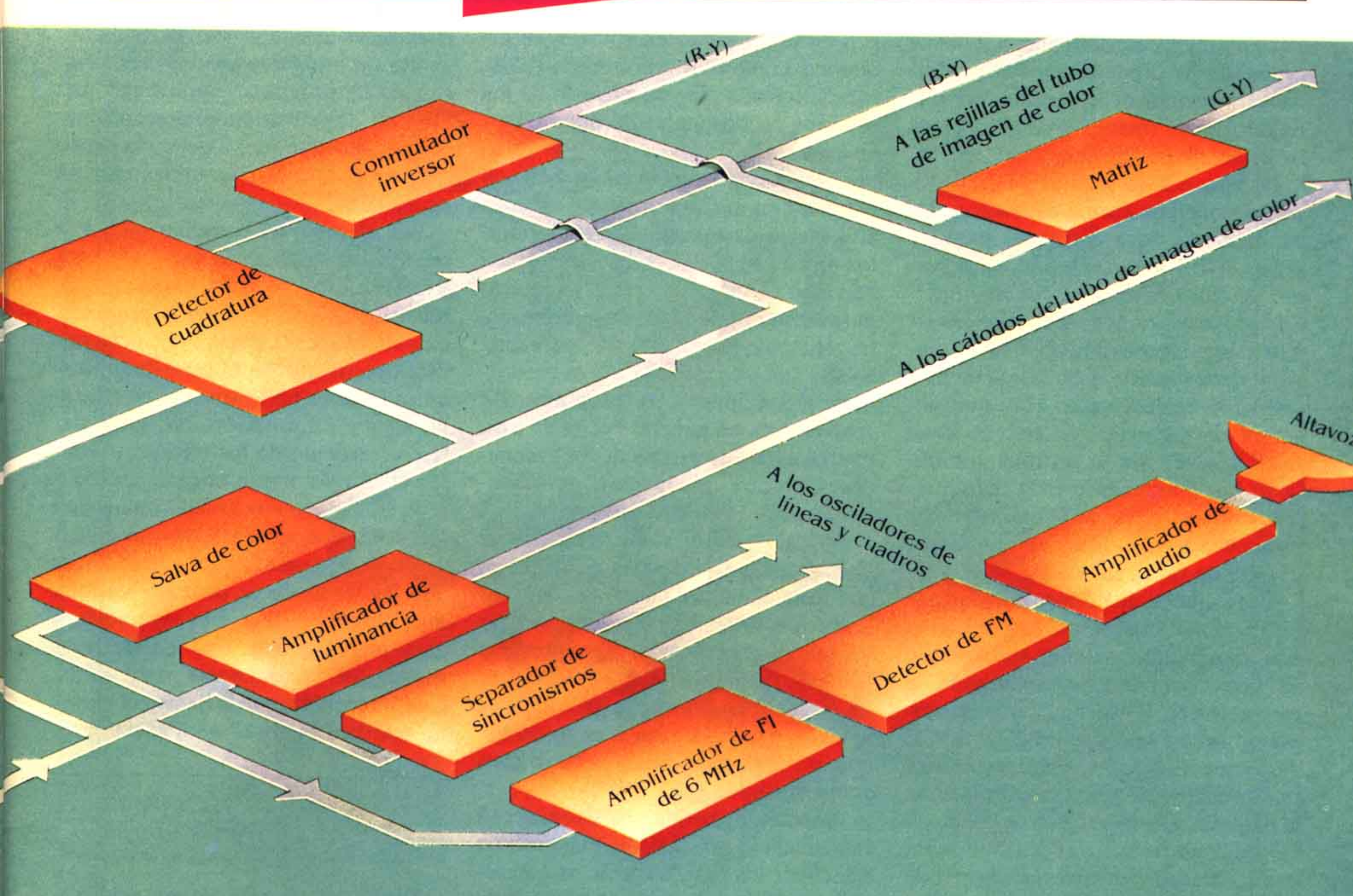
La señal compuesta de video constituye una especie de solución a medio camino entre la señal normal de televisión comercial y la entrada RGB. Sin embargo es mucho más aconsejable que la señal normal de televisión. Ello se debe que esta última señal sufre más procesos de modulación, conversión de frecuencia y amplificación hasta que se convierte en una imagen y es evidente que cuantos más procesos sufre una señal, más probable es que se vea contaminada por el ruido y la distorsión. En el modo RGB, la señal de sincronismo va por separado (en otra patilla del conector).

ANCHO DE BANDA

Existen unas razones técnicas interesantes aunque algo complicadas acerca del por qué un televisor no puede dar tan buena resolución como un monitor. Por ejemplo, un televisor no puede manejar información que requiera un ancho de banda en frecuencia superior a los 5,5 MHz. Con este ancho de banda se puede conseguir una presentación en pantalla de 40 columnas, pero para una presentación de 80 columnas que sea satisfactoria hace falta un ancho de banda superior a los 10 MHz.

Los monitores de video están especialmente diseñados para aceptar este ancho de banda. En casi todos los monitores disponibles en el mercado de ordenadores domésticos, la entrada prevista es la señal de video compuesto, por lo que no hace falta el sintonizador o el modulador que requieren los televisores. Debido a que necesita mucho menos procesamiento, la señal de vídeo compuesto de un ordenador aparece normalmente mucho más «limpia».

Cuando se habla de una «buena» imagen en la pantalla, normalmente se está haciendo referencia a la resolu-



ción y al color. Por desgracia ambas cosas no siempre son compatibles. Una pantalla de blanco y negro siempre tendrá una mejor definición de imagen que una pantalla en color. En una pantalla monocroma sólo se requiere un punto de fósforo para definir un punto de la imagen, mientras que se requieren tres puntos del mismo tamaño para definir ese mismo punto en una pantalla de color.

En una pantalla normal de cualquier clase hay unos 360.000 puntos de fósforo. En el caso de una pantalla de color se requerirían 120.000 de dichos puntos para el rojo, 120.000 para el verde y 120.000 para el azul. Esto significa que sólo hay disponibles 120.000 grupos de tres puntos para formar la imagen en una pantalla de color. Por el contrario una pantalla monocroma tiene 360.000 puntos disponibles para crear una imagen.

Todo el que haya trabajado con ambos tipos de pantalla, la monocroma y la de color, reconocerá inmediatamente la diferencia. El monitor en blanco y negro es mucho más nítido y claro que el de color, el cual presenta una imagen comparativamente más borrosa.

En muchas aplicaciones especializadas la elección natural será el monitor monocromo. El color normalmente elegido para estas aplicaciones —diseño asistido por ordenador o procesado de textos, por ejemplo— es el verde. Está considerado como el color más fácil para la vista y el menos cansado.

En cuanto a su peligrosidad para la salud, la inmensa mayoría de los informes que han sido publicados sobre el tema, declaran que el tubo de rayos catódicos es absolutamente inofensivo para los ojos. Normalmente se suelen oir quejas debidas a otras condiciones de trabajo, a colores de pantalla inadecuados, reverberaciones de la pantalla, distracción debida a otros focos luminosos y a veces efectos psicológicos. Sin embargo la pantalla de un televisor presenta un ligero centelleo y además suele estar hecha de un material reflectante que con frecuencia hace aparecer imágenes indeseadas y causantes de distracción. Esta es una de las razones por las que el uso de un televisor para un trabajo serio, que requiera una gran concentración durante largos períodos de tiempo, puede causar jaquecas.

Para producir una imagen con la misma resolución que las de la televisión o el vídeo ordinarios, nuestros ordenadores tendrían que controlar individualmente cada punto de fósforo de la pantalla. Y suponiendo que cada *pixel* necesite un byte de memoria, se necesitaría medio Megabyte para controlar toda la pantalla. Obviamente esto está fuera de lugar aquí, ya que

nos estamos ocupando de pequeños ordenadores domésticos y esa cantidad de memoria representa un tremendo gasto incluso en aplicaciones de ordenadores más grandes.

La mayoría de los micros domésticos disponen de aproximadamente 1K de memoria para dividir la pantalla entre 800 y 1000 cuadrados, típicamente 24 líneas por 40 columnas. Otros ordenadores te permiten elegir entre baja resolución y alta resolución, dependiendo de la forma de presentación seleccionada. Hay que observar sin embargo que esto no tiene nada que ver con la pantalla que utilices. Las limitaciones en la definición y la resolución de la pantalla vienen impuestas por el ordenador y no por el tubo de rayos catódicos.

Cada uno de estos aproximadamente 900 cuadrados es una casilla individual que puede contener un carácter. El ordenador contiene un conjunto de caracteres en ROM en el que corresponde un número del código ASCII para cada carácter. Cada carácter ocupa un cuadrado al ser presentado en la pantalla. De esta forma se puede almacenar un carácter en un solo byte, consiguiéndose un enorme ahorro de memoria.

HACIENDO UNA ELECCION

Los dos factores más importantes a tener en cuenta a la hora de comprar una pantalla nueva son el tipo de ordenador y las futuras aplicaciones. A pesar de ser un elemento importante dentro del sistema, la pantalla siempre ha de estar supeditada al ordenador. Si las capacidades gráficas del ordenador son pobres, seguirán siéndolo independientemente de lo buena que sea la pantalla. Siempre es el ordenador el que controla la pantalla y es importante que ésta se compre con arreglo a las características del ordenador.

También es importante tener en cuenta el uso a que se va a destinar el ordenador. En las aplicaciones en las que nunca va a ser necesario el color, la elección obligada es un monitor de blanco y negro. Si es esencial una imagen de alta calidad en colores, se requerirá un monitor con entrada RGB, si bien incluso para la mayoría de los usuarios profesionales de micros bastará con una entrada de vídeo compuesto. Para todos los poseedores de micros, con excepción de los de más alto precio, la elección entre un monitor de entrada RGB o uno de entrada de vídeo compuesto tendrá un interés puramente académico ya que se construyen muy pocos micros con salida RGB.

Para los juegos, la programación «recreativa» en general hecha en casa y otros usos domésticos de los ordenadores, bastará con un televisor ordinario. Aunque muchos juegos resultarían mucho mejor sobre pantallas de alta resolución, otros han sido escritos con conocimiento pleno de que sus gráficos sólo se iban a representar en un televisor ordinario. Ten presente además que debido a que los monitores se fabrican principalmente para aplicaciones comerciales, es frecuente que no incluyan características tales como el sonido.

Además, aunque su electrónica con frecuencia es más sencilla que la de un televisor, los monitores no son necesariamente más baratos que los televisores de un tamaño semejante. En lo que se refiere a pantallas, lo más grande no es necesariamente lo mejor. En muchos casos incluso podría ser mucho peor. Al aumentar o disminuir las dimensiones de la pantalla, lo que ocurre es que también aumenta o disminuye el tamaño de los *pixels* individuales. Si por ejemplo el ordenador produce 960 *pixels*, ése será el número de ellos que aparezcan en pantalla con independencia de lo grande o pequeña que ésta sea. Normalmente no se produce un aumento de la cantidad de información disponible en la pantalla al aumentar el tamaño de la misma, y en muchos casos lo que sucede es que una pantalla grande hace que ciertos contornos sean aún más destacados. Por cierto, recuerda que el tamaño de una pantalla se mide a lo largo de la diagonal de la misma, no según su anchura o su altura.

No se recomiendan los televisores viejos o los de muy bajo precio. Muchas veces lo que no parecería en absoluto un problema serio en una imagen de un programa emitido por una emisora de televisión comercial, puede convertirse en algo desastroso cuando el televisor se utiliza con un ordenador.

Por ejemplo el sobrebarrido, que recorta los lados de la parte superior e inferior de la pantalla, puede no parecer demasiado serio en la presentación de imágenes televisivas, pero podría ocurrir que en el caso de utilizar un ordenador no apareciera en pantalla alguna información útil. Otro defecto frecuente de los televisores viejos es el que ocurre cuando un color se presenta con mucho más intensidad que los otros, produciendo un efecto de emborronamiento que puede hacer que las letras de la pantalla resulten muy difíciles de leer. Las reparaciones para eliminar este tipo de defectos suelen ser normalmente muy caras. Con frecuencia es más barato comprar un televisor mejor... o un monitor.

CARACTERISTICAS DE LA VISUALIZACION

Tanto en el caso de un televisor como si tienes un monitor de video, puedes hacer varias cosas para mejorar las condiciones de visualización:

Puedes reducir la fatiga de la vista utilizando caracteres con colores claros contra un fondo negro. Para mejorar la definición utiliza los controles de brillo y contraste hasta lograr el ajuste óptimo. Los halos de color se pueden eliminar normalmente bajando la señal de color o eliminándola del todo; al fin y al cabo no necesitas el color para el procesamiento de textos.

Intenta situar tu mesa de trabajo, o el televisor o monitor, de forma que la pantalla no refleje la luz que entra por la ventana o la de alguna lámpara. En algunos casos puede ayudar el llevar ropas de tonos oscuros. Puedes minimizar aún más los efectos de las reflexiones teniendo la pantalla de forma que tengas que mirar ligeramente hacia abajo en vez de tenerla justamente enfrente de tus ojos o un poco más alta.

MODELIZANDO LA REALIDAD

- TIPOS DE MODELOS
- INCERTIDUMBRES
- DISTRIBUCION UNIFORME
- DISTRIBUCION EXPONENCIAL
- DISTRIBUCION NORMAL

Elegir la combinación ganadora en un juego de azar es equivalente a la extracción de números aleatorios. Es una tarea adecuada para tu micro, siempre que poseas unas nociones de estadística.

Los ingenieros de todas las especialidades construyen modelos, estructuras en miniatura, para probar sus diseños. Al hacer esto están simulando un sistema real. En el artículo de simulación, que presentamos hace unos meses, vimos que tu micro puede hacer algo semejante, como es ir recorriendo las etapas sucesivas de un determinado proceso para dar un resultado que se parezca a algo que ocurre realmente. Aquí te presentamos la manera de combinar los principios de la estadística y de la modelización para resolver problemas de una forma realista, tanto en juegos como en situaciones de la vida real.

TIPOS DE MODELOS

Técnicamente un modelo es una representación de un sistema real, que se construye por su sencillez y utilidad como ayuda en la toma de decisiones. Hay tres categorías de modelos: iconográficos, analógicos y simbólicos, aunque no todos resultan practicables en la simulación por parte de los entusiastas del micro doméstico.

Los modelos iconográficos no tienen partes móviles. A esta categoría pertenecen los mapas de carreteras y los aviones de juguete. Por ejemplo, en los estudios previos para la construcción de un puente sobre un estuario, los ingenieros de caminos podrían construir un modelo del puente y de la zona que rodea al estuario, para estudiar los efectos de la marea, las riadas y el viento sobre la estructura. Aunque se emplean ordenadores para monitorizar estos factores y analizar los resultados, el hecho de tener que recopilar datos a partir de ensayos reales significa que no hay necesidad de realizar una simulación.

En los modelos analógicos se representa una cantidad por otra. Por ejemplo el profesor **Phillips**, un economista británico, dispuso unos cuantos depósitos llenos de agua fría a diferentes alturas y utilizó el flujo de líquido entre los tanques para representar el flujo monetario en la economía del Reino Unido. Puede hacerse lo mismo electrónicamente, representando las diferentes cantidades por medio de diferentes valores de tensión eléctrica, pero para esto se emplean generalmente ordenadores analógicos, en lugar de los digitales, como es un micro doméstico. La tercera categoría, el modelo simbólico o abstracto, es la que resulta especialmente interesante para el que quiera llevar a cabo simulaciones en un ordenador doméstico.

INCERTIDUMBRES DE LA VIDA REAL

En los modelos de tipo simbólico, se utiliza una fórmula matemática para representar el sistema real. Por ejemplo, si quieres estudiar la distancia (S) recorrida por un coche que viaja a velocidad constante (V) durante un cierto tiempo (T), el modelo adecuado sería S = V*T. Este tipo de



ecuación se llama determinística, debido a que no da ninguna medida de las incertidumbres. Describe con precisión el movimiento del automóvil para cualquier conjunto de valores que especifiques. Sin embargo existen muchos sucesos que no pueden ser especificados sin un cierto grado de incertidumbre. Tienes un ejemplo de esto en el lanzamiento al aire de una moneda. Los modelos que describen este tipo de sucesos imprecisos se llaman estocásticos. Normalmente son este tipo de sucesos los que requieren simulación, particularmente en los juegos.

Algunas variables pueden tomar solamente valores discretos, tales como 0, 1, 2, 3, ... A esta categoría pertenecen el número de goles en un partido de fútbol o el número de hijos varones de una familia. Cuando se simulan variables aleatorias de este tipo, es importante darse cuenta de que existen muchos procesos aparentemente diferentes que tienen de hecho la misma estructura subyacente.



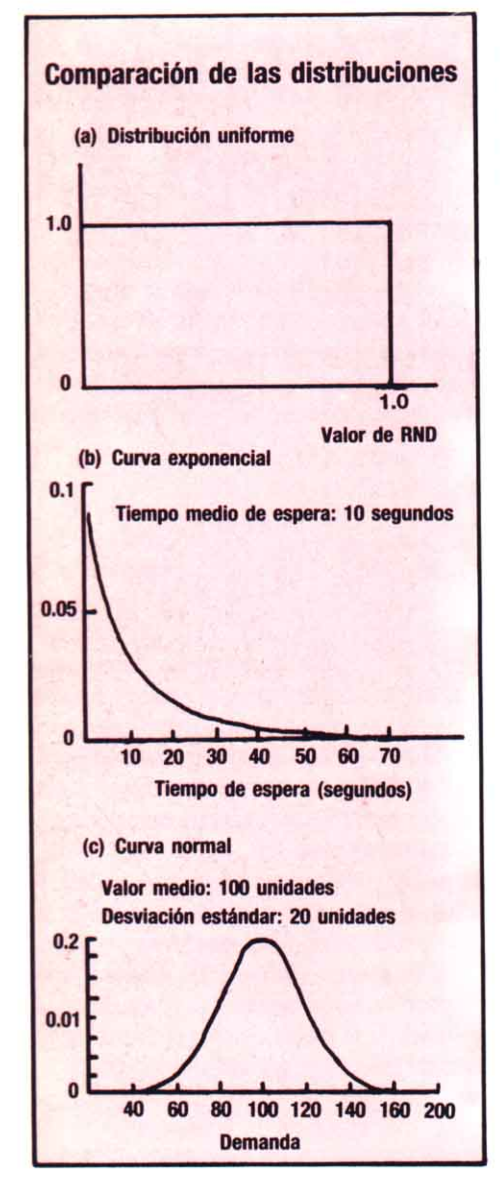
Consideremos por ejemplo el caso de la simulación del lanzamiento de una moneda al aire, anteriormente mencionado. Este programa podría modificarse fácilmente para representar los resultados aleatorios de una partida de dados o el número de goles conseguidos en una serie de lanzamientos de *penalties*. En todos estos casos hay un cierto número de pruebas repetidas independientes, cuyos resultados pueden producirse con probabilidades bien definidas. Supongamos que la prueba consiste en lanzar una moneda al aire cuatro veces. El resultado de cada prueba puede ser cuatro caras, tres caras y una cruz, dos caras y dos cruces, una cara y tres cruces o cuatro cruces. Además hay una probabilidad bien definida asociada con cada uno de estos posibles resultados. Las estructuras de este tipo, llamadas procesos de **Bernoulli**, generan variables aleatorias discretas que no resultan adecuadas, por ejemplo, para el estudio del número de accidentes que ocurren en un **Grand Prix**, el número total de llamadas telefónicas que se producen en un período de media hora o el número de rayos registrado durante una tormenta eléctrica. En todos estos ejemplos, los sucesos se producen de modo casual durante un determinado período de tiempo, ellos solos y no como consecuencia de un conjunto de pruebas repetidas. A estas estructuras se les llama procesos de **Poisson**. Teclea el siguiente programa que te permite realizar un muestreo a partir de cualquier distribución de **Poisson** que elijas:

```
10 KEY OFF:CLS:R=RND(-TIME)
20 LOCATE 8,9:PRINT
   "SIMULACION DE POISSON"
   :LOCATE 8,10:PRINT"=======
   == == ======="
50 LOCATE 5,14:INPUT"Cual es
   el valor medio";A
60 LOCATE 4,2:PRINT"VALOR
   MEDIO":LOCATE 7,3
   :PRINT A
65 LOCATE 5,14:PRINT SPC(30):
   LOCATE5,14:INPUT"Cuantas
   muestras";N:LOCATE 5,14
   :PRINT SPC(30)
70 LOCATE 20,2:PRINT
   "NO. MUESTRAS":LOCATE
   24,3:PRINT N:LOCATE 7,14
   :PRINT"ESPERA":FOR J=1
   TO 1000:NEXT:CLS
75 LOCATE5,2:PRINT"MUESTRAS"
80 FOR I=1 TO N
90 C=0:T=1
100 S=EXP(-A)
110 T=T*RND(1)
120 IF T<= S THEN PRINT
    TAB(16) C:GOTO150
130 C=C+1
140 GOTO 110
150 NEXT I
160 LOCATE 0,22:PRINT
    "Otra vez (s/n)?"
170 A$=INKEY$:IF A$=""
    THEN 170 ELSE IF A$="s"
    THEN GOTO 10
180 CLS:END
```

Al ejecutar el programa, se te pide que introduzcas un valor promedio (la media aritmética) y el tamaño de tu muestra (líneas 50 y 70). La parte principal del programa (líneas 80 a 150) utiliza una ecuación para generar las variables de **Poisson**. A la variable S se le asigna (línea 100) el valor del número e (una constante matemática) elevado a la potencia A (el valor de la media que tú has introducido). La línea 110 elige un número aleatorio entre 0 y el valor de T, y realiza un cambio de escala según el valor de T. La línea 120 compara S y T para decidir si hay que presentar o no una variable (C).

Supongamos por ejemplo que estás diseñando un juego espacial en el que una nave interplanetaria tiene que atravesar una tormenta de meteoritos. Si el número medio de «impactos» en cualquier período de un minuto es dos y se requieren cinco simulaciones separadas de un minuto cada una, el ordenador podría presentar los números 2, 3, 2, 1, 0 como estadística de impactos. Cada elemento de la lista no será muy diferente de 2 (la media), y en total hay cinco elementos (el tamaño de la muestra).

¿Podría utilizarse esta misma técnica para simular, digamos, el tanteo final de un partido de fútbol? Está cla-



ro que se trata de una variable discreta, ya que el número de goles conseguidos sólo puede tomar valores enteros: 0, 1, 2, 3, etc. Además son sucesos que se producen por sí solos en el tiempo y no como resultado de pruebas repetidas. En consecuencia se trata de un caso susceptible de ser adecuadamente descrito mediante un proceso de **Poisson**. Aparte del número de partidos jugados, la única información complementaria que necesitas en este caso es qué valor hay que utilizar como número medio de goles. En este punto se hace necesario salir al mundo real y recoger algunos datos.

Tomando como base los goles marcados en y fuera de casa durante la Liga una elección razonable para los tanteos sería adoptar el valor 1.7 como media de goles domésticos y 1.0 como media de goles fuera de casa. La división por dos de cada uno de estos valores conduce a un tanteo medio por cada medio partido de 0.85 y 0.5. La construcción del programa resulta ya inmediata:

```
20 DIM A$(25),H$(25),H(50)
   ,A(50),FA(25),FH(25)
   :R=RND(-TIME)
50 KEY OFF:CLS:LOCATE 9,2
   :PRINT"LA LIGA DE
   POISSON"
70 LOCATE 5,10:INPUT"Cuantos
   encuentros (1-25)";N
75 IF N<1 OR N>25 THEN 70
77 CLS:LOCATE 5,2:PRINT
   "NOMBRES DE LOS EQUIPOS"
   :LOCATE 0,4:PRINT"PARTIDO
   E. DE CASA  E. DE FUERA"
80 FOR I=1 TO N
90 PRINT TAB(2);I;
100 PRINT TAB(9);"";
    :INPUT H$(I)
110 PRINT TAB(21);"";
    :INPUT A$(I)
120 NEXT I
130 FOR V=1 TO 1000:NEXT V
150 FOR I=1 TO 2*N
160 C=0:T=1
170 S=EXP(-.85)
180 T=T*RND(1)
190 IF T<=S THEN H(I)=C
    :GOTO220
200 C=C+1
210 GOTO 180
220 NEXT I
230 FOR I=1 TO 2*N
240 C=0:T=1
250 S=EXP(-.5)
260 T=T*RND(1)
270 IF T<=S THEN A(I)=C
    :GOTO300
280 C=C+1
290 GOTO 260
300 NEXT I
320 CLS:PRINT"RESULTADOS
    A LA MITAD DEL PARTIDO"
330 FOR I=1 TO N
340 PRINT H$(I);H(I)
    :PRINT A$(I);A(I)
    :PRINT
350 FOR D=1 TO 200:NEXT D,I
360 PRINT"PULSA UNA TECLA"
370 T$=INKEY$:IF T$=""
    THEN 370
380 CLS:PRINT"RESULTADOS
    FINALES"
390 FOR I=1 TO N
400 FH(I)=H(I)+H(N+1)
410 FA(I)=A(I)+A(I+N)
420 PRINT H$(I);FH(I)
    :PRINT A$(I);FA(I)
    :PRINT
430 FOR D=1 TO 200:NEXT D,I
440 PRINT"Otra vez (s/n)?"
450 T$=INKEY$:IF T$=""
    THEN 450
455 IF T$="n" THEN CLS:END
460 PRINT"Los mismos equipos
    (s/n)?"
470 T$=INKEY$:IF T$=""
    THEN 470 ELSE IF T$="n"
    THEN RUN
480 IF T$="s" THEN GOTO
    150 ELSE GOTO 470
```

Ejecuta el programa y responde a las preguntas que te hace su primera sección (desde el principio hasta la lí-



nea 120). Para evitar el tener que teclear demasiado, puedes introducir letras del alfabeto en vez de nombres reales para ambos equipos. Las líneas 150 a 300 utilizan el algoritmo de **Poisson** que vimos en el primer programa, para generar los tanteos del medio tiempo en casa y fuera de casa. Las líneas 320 a 350 organizan la presentación de las puntuaciones de medio tiempo y las líneas 390 a 430 se ocupan del resultado de los partidos completos.

Aunque los goles individuales simulados por partido no correspondan con los resultados reales, el comportamiento global de la situación simulada debería parecerse bastante a dichos resultados reales.

La distribución de **Poisson** funciona bien con variables que tomen valores enteros; pero hay otras variables aleatorias continuas que toman valores fraccionarios. Por ejemplo en una simulación de ordenador del salto de longitud en una **Olimpiada**, un participante podría saltar una distancia cualquiera entre seis y nueve metros. La longitud de dicho salto es un ejemplo de variable aleatoria continua.

Aunque la función RND del ordenador es adecuada para la simulación de variables aleatorias discretas de tipo **Poisson**, necesitas una función RND modificada para la modelización con variables continuas. La figura 1 compara el tipo de distribución dado por ambos tipos de variables. Las variables discretas dan una distribución uniforme, mientras que las variables continuas pueden dar una distribución exponencial o normal. Teclea el siguiente programa para transformar las variables aleatorias planas o uniformemente distribuídas que te da la función RND de tu ordenador en variables aleatorias con distribución exponencial.

```
40 KEY OFF:CLS:R=RND(-TIME)
   :LOCATE 5,2:PRINT
   "SIMULACION EXPONENCIAL"
50 LOCATE 2,10:INPUT
   "Cual es el valor medio"
   ;A
60 LOCATE 2,10:PRINT SPC(30)
70 LOCATE 2,10:INPUT
   "Numero de muestras";N
75 LOCATE 2,10:PRINT SPC(30)
   :FORJ=1 TO 1000:NEXT
   :CLS:PRINT"Resultados
   de la simulacion":PRINT
80 FOR I=1 TO N
90 X=(-A)*LOG(RND(1))
   :Y=INT(X*10):PRINT.1*Y
100 NEXT I
110 LOCATE0,20:PRINT
    "Otra vez (s/n)?"
120 T$=INKEY$:IF T$="s"
    THEN RUN
130 IF T$<>"n" THEN 120
140 CLS:END
```



Al ejecutar este programa tienes que especificar el valor medio y el número de muestras requeridas; a continuación aparecerán en pantalla las variables. Todo se hace en la línea 90. Se utiliza una técnica matemática llamada método de la transformada inversa, para convertir variables «planas» en variables exponenciales. Este programa puede usarse también perfectamente para simular, por ejemplo, el tiempo de espera de una secadora en una lavandería, o la vida media de los paquetes de energía en un juego de aventura espacial. En todos los casos el concepto importante es el del tiempo transcurrido.

DISTRIBUCION NORMAL

La curva normal en forma de campana representa la más famosa de las distribuciones estadísticas. Ha resultado muy útil en la descripción de todo tipo de fenómenos naturales, incluyendo las tallas y los pesos de las personas adultas. Además está ampliamente aceptada como el medio más apropiado de describir la distribución de errores.

Para ilustrar la forma en que podría incorporarse la distribución normal dentro de un modelo, consideremos el siguiente ejemplo para diseñar un juego comercial. Supongamos que las ventas (S) de una empresa dependen de algún nivel base (S0) y de la cantidad que se ha gastado en publicidad



(A). Esta relación podría resumirse mediante le expresión S = S0 + (B*A), siendo B una constante conocida.

Pero en la vida real raramente se dan estas relaciones tan precisas. Los sucesos y circunstancias imprevistos distorsionan los hechos o introducen «ruido» en el sistema. Este sistema se puede modelizar adecuadamente introduciendo una ecuación de perturbación con distribución normal para las ventas. La ecuación completa para la cifra de ventas sería entonces S = S0 + (B*A) + Distribución normal.

Para ver cómo funciona, teclea el siguiente programa:

```
10 CLS:KEY OFF
   :R=RND(-TIME)
40 DIM U(50),Y(50),Z(50)
50 LOCATE 5,2:PRINT
   "SIMULACION NORMAL"
60 LOCATE 2,10:INPUT"Cual es
   el valor medio";A
65 LOCATE 2,10:PRINT SPC(30)
70 LOCATE 2,10:INPUT
   "Cual el el valor minimo"
   ;M
75 LOCATE 2,10:PRINT SPC(30)
80 LOCATE 2,10:INPUT
   "Cuantas muestras";N
   :FOR J=1 TO 1000:NEXT:CLS:
   PRINT"Resultados de la
   simulacion":PRINT
90 B=(A-M)/2.5
100 FOR I=1 TO N
110 T=0
120 FOR J=1 TO 15
130 Y(J)=RND(1)
140 T=T+Y(J):NEXT J
150 U(I)=((T/15)-.5)/SQR
    (1/(12*15))
160 Z(I)=A+(B*U(I))
170 P=INT(10*Z(I)):PRINT.1*P
180 NEXT I
190 LOCATE 0,20:PRINT
    "Otra vez (s/n)?"
200 T$=INKEY$:IF T$="s"
    THEN RUN
210 IF T$<>"n" THEN 200
220 CLS:END
```

Ejecuta la simulación y especifica los valores medio y mínimo de tu variable. Técnicamente, el margen de valores de la curva normal no está acotado, por lo que existe una pequeña probabilidad de que un valor simulado sea menor que el valor mínimo especificado por tí. Pero esto no sucederá muy a menudo.

Las líneas 120 a 150 utilizan la función RND para generar 15 números aleatorios entre 0 y 1 y a continuación sumarlos. La línea 160 calcula la media y le añade un factor de escala. Después de otro cambio de escala (línea 160) se puede imprimir el resultado.

Tal vez te ayude a tener una sensación más real de lo que ocurre, el hecho de hacer que los números se generen a partir de dos distribuciones con la misma media y aproximadamente el mismo margen de valores, pero con diferentes formas. Ejecuta este programa introduciendo los valores de 100 para la media, 50 para el valor mínimo y 40 para el tamaño de la muestra. Con los valores obtenidos puedes hacer un gráfico parecido al de la figura 1C.

Borra ahora la línea 90 y las líneas 120 a 160, y escribe de nuevo la siguiente línea:

```
110 Z(I)=M+RND(1)*2*(A-M)
```

Estos cambios hacen que el programa sea capaz de presentar valores aleatorios con distribución uniforme. Ejecuta de nuevo el programa, introduciendo los mismos valores que antes y compara los resultados. Al dibujar esta vez los valores se obtiene un gráfico como el de la figura 1B. Observa que las lecturas normales (las de la primera prueba) están mucho más apiñadas en torno al valor medio.

CREACION DE NIVELES DE DIFICULTAD

- UN LABERINTO ALEATORIO
- DOS MANERAS DE CONVERTIR EL JUEGO EN MAS DIFICIL
- COMO MOVER AL JUGADOR
- AÑADIENDO LA PUNTUACION

Algunos juegos de laberintos son muy fáciles de resolver, pero éste te dará dos niveles de dificultad y además presentará cada vez un laberinto diferente. Busca la manera de atravesarlo para encontrar el tesoro.

Los juegos de ordenador con frecuencia te piden que selecciones un nivel de dificultad antes de empezar a jugar. Esto hace posible que tanto principiantes como expertos puedan practicar el mismo juego, sin que sea demasiado difícil ni demasiado fácil para nadie.

Dependiendo de la naturaleza del juego, tienes muchas maneras de introducir el nivel de dificultad. Por ejemplo, puedes cambiar el número de enemigos, introducir una serie de retardos en el juego, permitir un tiempo mayor o menor, cambiar los obstáculos, etc.

En nuestro coleccionable ahora puedes ver cuántos niveles de dificultad se pueden introducir en un juego de laberintos. El juego utiliza una o dos formas de generar niveles de dificultad distintos. No se trata tan sólo de encontrar un camino para recorrer el laberinto, sino que además el jugador dispone de un tiempo fijo limitado en el que guiar a un hombre hasta algún tesoro que aparece dibujado en alguna parte del laberinto.

Para que la cosa resulte más difícil, puedes utilizar dos métodos. El primero consiste en cambiar la complejidad del laberinto. El otro es alterar el tiempo límite.

VIDAS

Cuando al jugador que está intentando alcanzar el tesoro, se le termine el tiempo, querrás imponerle algún tipo de penalización. Podrías hacer que el jugador perdiera algo de su puntuación, pero la penalización más ampliamente usada es hacer que pierda una vida.

Aqui el jugador recibe tres vidas, por lo que si no consigue encontrar el tesoro dentro del tiempo límite e tres ocasiones seguidas, el juego termina-

LABERINTOS ALEATORIOS

El juego de laberintos está basado en una subrutina de generación aleará.



toria de laberintos, que resulta un programa interesante por sí mismo, ya que dibuja cada vez un laberinto diferente, evitándote el tener que crear toda una serie de laberintos. En la página 14 vimos la forma de generar un laberinto con sentencias DATA y cómo incorporarlo en un programa; imagínate lo complicado que sería el tener que generar toda una serie de ellos.

El diseño de laberintos aleatorios es mucho más fácil que eso, pero más complicado de lo que te puedes imaginar. Una forma obvia de diseñarlos podría ser imprimir un número de bloques, por ejemplo gráficos incluidos en la ROM, aleatoriamente sobre la pantalla. Pero el problema es que podría resultar que no se obtuviera un laberinto, ya que no se garantiza que haya un camino a través del mismo; por ello, para usar este método habría que introducir alguna forma de comprobar que existe una salida.

COMO DIBUJAR LABERINTOS ALEATORIOS

La mejor manera de dibujar laberintos aleatorios es hacer un programa que dibuje una trayectoria aleatoria, y disponer la misma en forma de laberinto. El programa de tu máquina está diseñado de forma que la línea está contenida dentro de una trama dibujada sobre la pantalla. No se permite que la línea se cruce consigo misma en ningún caso. Cuando la trayectoria aleatoria ya no puede avanzar más —bien porque se encuentra con una esquina, o entre ella misma y la trama, o incluso puede quedar atrapada dentro de sí misma— el ordenador vuelve sobre sus pasos. Esto lo hace retrocediendo un paso cada vez y examinando la zona de alrededor. Cuando la máquina la encuentra, se inicia una nueva rama de la trayectoria aleatoria, por la cual continúa hasta que se tropieza de nuevo y empieza a retrasar sus pasos otra vez. El ordenador sigue intentando dibujar nuevas ramas hasta que la trama está llena, en cuyo caso vuelve al sitio donde empezó.

Después de que el programa ha terminado de dibujar el laberinto, sólo hay un camino posible para recorrerlo, que puede resultar muy fácil, ya que las ramas del camino no son complicadas. El laberinto también se puede recorrer con la «regla de la mano derecha» siguiendo la pared derecha (o la izquierda, según los casos) del laberinto durante todo el tiempo. Para evitar que alguien pueda hacer esto, te hacen falta «islas» en el laberinto, con las que interrumpir las paredes. Así, después de dibujar el laberinto, el programa dibuja una serie de bloques aleatorios que hacen que el laberinto parezca más complicado y que desconcertarán al que pretenda servirse de la regla de la mano derecha.

Cuando hayas introducido el programa completo, almacénalo (con SAVE) ya que en el siguiente artículo veremos la manera de añadirle algunos efectos sonoros.

Al ejecutar este programa (RUN) se te pide que elijas un nivel de dificultad. Puedes seleccionar un número entre 1, 2, 3 ó 4, que corresponden a duraciones de juego de 32, 24, 16 y 8 segundos. El objetivo del juego es alcanzar el tesoro, un asterisco situado al azar, en el menor tiempo posible. Tu «hombre» es un carácter # y para dirigirle hacia el blanco se utilizan las teclas de cursor.

```
39 'nivel de dificultad
40 SCREEN 0:CLS:WIDTH35
   :INPUT"Nivel de
   dificultad? (1-4) ";A
   :IF A<1 OR A>4
   THEN 40
50 SCREEN 1:CLS:WIDTH 30
   :TL=5-A:TL=TL*400:VI=3
   :DIM A(4)
69 'dibujo del laberinto
70 R=RND(-TIME):CLS:
   VPOKE BASE(6),&H44
   :KEY OFF
80 CO=BASE(5)
85 FOR J=0 TO 20:FI=CO+32*J
90 FOR N=FI+2 TO FI+30
   :VPOKE N,219:NEXT N
95 NEXT J
100 B=BASE(5)+49:A=B
   :VPOKE A,5
110 A(1)=-1:A(2)=-32
   :A(3)=1:A(4)=32
130 J=INT(RND(1)*4)+1:G=J
140 B=A+A(J)*2:IF VPEEK(B)
   =219 AND VPEEK(A+A(J))
   =219 THEN VPOKEB,J
   :VPOKE A+A(J),32:A=B
   :GOTO130
150 J=(J+1):IF J=5 THEN J=1
160 IF J<>G THEN GOTO 140
170 A=A-A(VPEEK(A))*2
   :IF VPEEK(A)=5 GOTO
   1000 ELSE GOTO130
999 'puntos aleatorios
1000 FOR Z=1 TO 10
1002 X=INT((RND(1)*19)+1)
   *32+INT(RND(1)*25)+4
1004 IF VPEEK(CO+X)=219
   THEN VPOKECO+X,32ELSE
   GOTO1002
1005 NEXT Z
1006 'colocacion tesoro
1007 X=(INT(RND(1)*20)+1)*
   32+(INT(RND(1)*27)+3)
   :IF VPEEK(CO+X)=219
   THEN 1007
1010 TE=CO+X:VPOKE TE,42
1012 TIME=0
1014 H=BASE(5)+49:HA=H
   :LOCATE 0,21:PRINT"PM"
   ;PM
1016 LOCATE 0,22:PRINT"VI"
   ;VI;" TI";:PRINT USING
   "#####";TIME/50;
   :PRINT" PU  ";PU
1018 IF TIME>TL THEN 2000
1020 'lectura teclado
1024 A=STICK(0)
1026 IF A=1 THEN H=HA-32
   :GOTO1034
1028 IF A=3 THEN H=HA+1
   :GOTO1034
1030 IF A=5 THEN H=HA+32
   :GOTO1034
1032 IF A=7 THEN H=HA-1
   :GOTO1034
1034 IF VPEEK(H)=42 THEN
   VPOKEH,32:VPOKE HA,32
   :GOTO 3000
1036 IF VPEEK(H)<>219 THEN
   VPOKE HA,32:VPOKEH,35
   :HA=H
1038 GOTO 1016
2000 VI=VI-1:FOR Z=1 TO
   100:VPOKE HA,Z:NEXT
```

```
     :BEEP:VPOKEHA,32
2002 IF VI>0 THEN 1012
2003 FOR J=1 TO 10:BEEP
     :NEXT:VPOKE TE,32:PU=0
2004 LOCATE0,22:PRINT
     "Otra vez (s/n)
     ":R$=INKEY$:IF R$=""
     THEN 2004
2006 IF R$="s" THEN 2012
2008 IF R$="n" THEN SCREEN 0
     :LOCATE 0,0:PRINT
     "Programa terminado"
     :END
2010 GOTO 2004
2012 LOCATE0,22:PRINT
     "Otro laberinto? (s/n)
     ":R$=INKEY$:IF R$=""
     THEN 2012
2014 IF R$="s" THEN VI=3
     :GOTO 70
2016 IF R$="n" THEN VI=3
     :GOTO 1006
2018 GOTO2012
3000 PU=INT(PU+100-TIME/50)
     :IF PU>PM THEN PM=PU
3010 GOTO 1006
```

El programa comienza por pedirnos el nivel de dificultad del juego. Dicho nivel, que se introduce en la línea 40, determina el valor del tiempo límite de cada vida, mediante la variable TL de la línea 50. La relación es inversa, es decir, cuanto más pequeño es el nivel de dificultad, mayor es el tiempo límite disponible. Además, en estas líneas se asigna el modo de pantalla SCREEN 1, se fija en tres el número de vidas (variable VI) y se dimensiona la matriz A que utilizaremos más adelante.

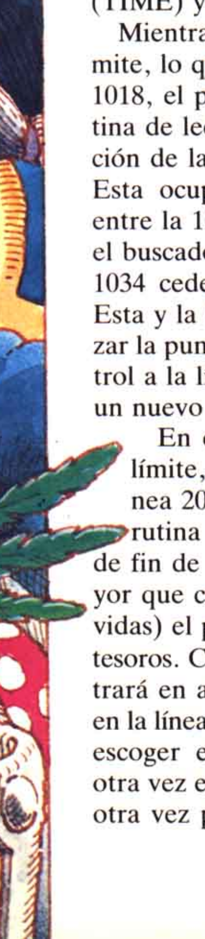
La verdadera rutina de creación del laberinto ocupa las líneas que van de la 69 a la 170. Vamos a comentar a fondo esta rutina, ya que hace uso de algunas interesantes posibilidades del ordenador, al trabajar directamente con la VRAM. La rutina comienza en la línea 70 limpiando la pantalla, eliminando la línea de teclas de función e introduciendo el valor &H44 en el primer byte de la tabla 6. Esta tabla controla los colores de los caracteres en el modo SCREEN 1. Al introducir el valor &H44 en el primer byte de la tabla 6, conseguimos que los caracteres correspondientes a los códigos 0 a 7, aparezcan en la pantalla en color azul sobre fondo azul, es decir, no se van a ver, pero aunque no se vean van a estar ahí y nos van a ayudar a construir el laberinto. A continuación, entre las líneas 80 y 95, se dibuja el fondo del laberinto, un rectángulo de 21 filas × 28 columnas. Este rectángulo se dibuja POKEando el valor 219 (que corresponde a un cuadrado) en la zona de VRAM correspondiente a la pantalla, es decir, a partir de la dirección de comienzo de la tabla 5 (BASE(5)). Si quieres experimentar, puedes cambiar los límites de los bucles FOR...NEXT para obtener un laberinto más grande o más pequeño, puedes también cambiar el valor 219 por otro, lo que hará que cambien los caracteres que definen el fondo y, por último, puedes cambiar de color actuando sobre el byte correspondiente de la tabla 6. Recuerda para ello que cada uno de los 32 bytes de la tabla 6 determina el color de 8 caracteres (el byte cero de los caracteres de códigos 0 a 7, el 1 de los de códigos 8 a 15, etc).

Una vez dibujado el fondo, se elige la posición de inicio del laberinto (en la línea 100). Esta posición corresponde a la dirección BASE(5) + 49 de la VRAM. Si quieres cambiarla no tienes más que escoger un valor distinto de 49. En dicha posición se POKEa el caracter de código 5, pero como hemos dicho que aparece en azul y sobre fondo azul, no podremos verlo. En la línea 110 se definen los 4 posibles movimientos de la traza del laberinto. A(1) y A(3) corresponden a movimientos a izquierda y derecha respectivamente, mientras que A(2) y A(4) representan las direcciones arriba y abajo.

Las líneas comprendidas entre la 130 y la 170 dibujan el laberinto aleatorio de la siguiente forma: primero, en la línea 130, se elige una dirección

aleatoria entre 1 y 4 (arriba,abajo, izquierda o derecha). En dicha dirección vamos a intentar dibujar el laberinto. Para ello comprobamos, en la línea 140, si podemos avanzar 2 caracteres en la dirección elegida. Si en las dos posiciones, en la dirección considerada, nos encontramos con el código 219, quiere decir que estamos sobre el fondo del laberinto y que podemos dibujar la traza. En este caso imprimimos un espacio (código 32) en el primer caracter, mientras que en el segundo, y aquí está el truco, imprimimos la dirección de movimiento (1,2,3 o 4). Como esta dirección se imprime en azul sobre fondo azul, no se ve, pero está ahí y nos va a permitir volver sobre nuestros pasos, retrocediendo a través del laberinto. En el caso de que no podamos avanzar en la dirección elegida, al salirnos de los límites del laberinto o bien porque haríamos que la traza se cruzara a sí misma, se elige la siguiente dirección (en la línea 150) y se intenta de nuevo (a través del salto a la 140, desde la línea 160).

Cuando se hayan comprobado las 4 posibles direcciones y no haya sido posible avanzar, pasaremos a la línea 170. Ella se encarga de hacernos retroceder. Para ello y recordando que habíamos dejado escrita, en cada punto, la dirección por la que veníamos, no hay más que moverse en la dirección contraria. De esta forma se retroceden 2 posiciones en el laberinto y se salta de nuevo a la línea 130, para continuar con el laberinto en otra dirección. Cuando el laberinto esté terminado, iremos retrocediendo, posición a posición, hasta llegar a la posición de partida. La línea 170 detectará la llegada a la posición inicial ya que el VPEEK de esta posición es 5 (lo habíamos escrito para este fin en la línea 100). Así pues, la línea 170 detectará que se ha terminado de dibujar el laberinto y dará paso a la siguiente parte del programa que comienza en la línea 1000. Esta y las siguientes, hasta la 1005, se ocupan de colocar aleatoriamente 10 espacios sobre el laberinto, estableciendo así 10 pasos o túneles a través de las paredes del mismo. Si quieres que aparezcan más o menos, no tienes más que cambiar el valor 10, en la línea 1000, por otro valor. Incluso puedes hacer que el número dependa del nivel de dificultad elegido, por ejemplo, a mayor nivel de dificultad, más túneles. Para ello tendrás que cambiar el valor 10 de la línea 1000 por una variable cualquiera que tomará su valor en función del nivel de dificultad elegido. Siguiendo con el programa, las líneas que van de la 1006 a la 1010, se ocupan de elegir una posición aleatoria en la que que colocar el tesoro (representado por un asterisco, de código 42).



A partir de este momento comienza la cuenta atrás. El programa, en las líneas 1012 a 1016, pone a cero el reloj interno y se ocupa de imprimir, en las 2 líneas inferiores de la pantalla, los valores de puntuación máxima (PM), vidas (VI), tiempo transcurrido (TIME) y puntuación (PU).

Mientras no se alcance el tiempo límite, lo que se comprueba en la línea 1018, el programa trabajará en la rutina de lectura de teclado y actualización de la posición del busca tesoros. Esta ocupa las líneas comprendidas entre la 1020 y la 1038. Cada vez que el buscador alcance el tesoro, la línea 1034 cederá control a la línea 3000. Esta y la 3010 se encargan de actualizar la puntuación, devolviendo el control a la línea 1006 para que aparezca un nuevo tesoro.

En caso de cumplirse el tiempo límite, el programa se dirige a la línea 2000. En esta línea se inicia la rutina de comprobación de vidas y de fin de juego. Mientras VI sea mayor que cero (es decir mientras haya vidas) el programa seguirá ofreciendo tesoros. Cuando se acaben las vidas entrará en acción la rutina que empieza en la línea 2003 encargada de hacernos, escoger entre dejar de jugar, jugar otra vez en el mismo laberinto o jugar otra vez pero en un laberinto nuevo.

ROMPIENDO LA BARRERA DEL SONIDO

- CREANDO EL SONIDO CORRECTO
- EFECTOS ESPECIALES PARA EL LABERINTO
- RUIDOS DE EXPLOSIONES, SILBIDOS Y DISPAROS

Puedes darles vida a tus juegos de ordenador y hacerlos más excitantes añadiéndoles algunos efectos sonoros; vale todo, desde pitidos, explosiones y el ruido de los disparos de los invasores, hasta una marcha fúnebre o un tren de vapor.

Normalmente los programas de juegos suelen llevar incorporados toda clase de efectos sonoros que los hacen más excitantes: explosiones, disparos, pequeñas melodías o cualquier otra cosa que permita la imaginación del programador.

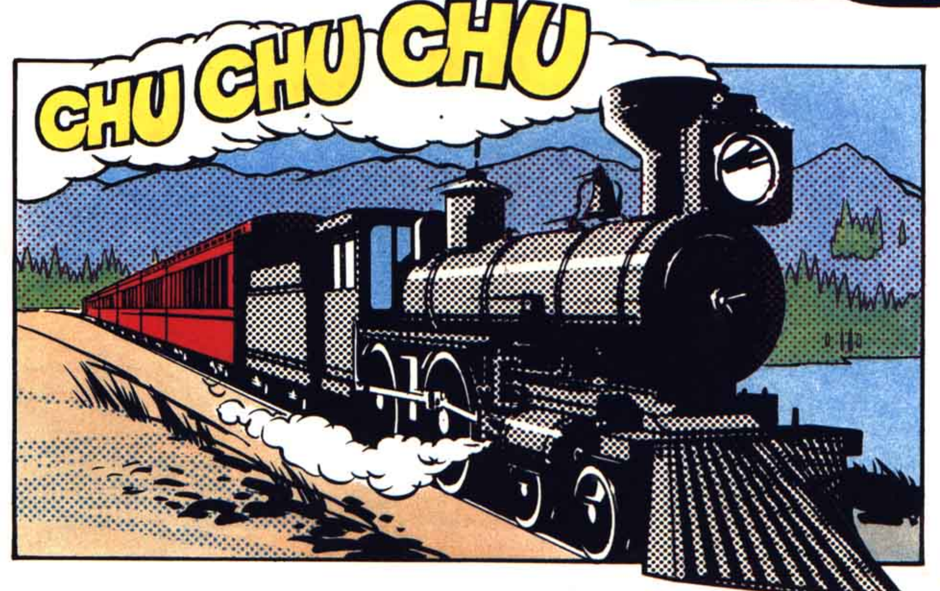
En este capítulo de nuestro coleccionable te presentamos un pequeño repertorio de efectos sonoros prefabricados, que puedes utilizar tal como vienen aquí o como base para nuevos experimentos.

Recuerda que no existen reglas infalibles y recetas rápidas para producción de efectos sonoros. Si lo que tu juego necesita es un sonido para acompañar una escena en la que se ve a un sujeto al que están zurrando, no tendrás más remedio que sentarte ante tu máquina y hacer diversas pruebas. Recíprocamente, un sonido aparentemente sin sentido puede resultar magnífico si le encuentras el conjunto de gráficos adecuados.

La forma de incorporar los efectos sonoros en tus programas depende de su complejidad y de la frecuencia con que los vayas a utilizar. Para efectos sencillos que sólo se utilizan una vez en el programa, puede ser suficiente una estructura del tipo IF ... THEN, pero para los más complejos o que se utilizan varias veces, es mejor escribir una subrutina. La sofisticación de los efectos que puedas conseguir, depende parcialmente de tu propia habilidad de programación, pero también está determinada por las capacidades sonoras de tu ordenador. Por ejemplo tu **MSX** tiene un generador de sonidos, capaz de sintetizar una enorme cantidad de efectos diferentes.

El generador de sonidos es extremadamente sofisticado y dispone de tres «voces» o canales de sonido fácilmente controlables. Se puede conseguir una gran variedad de sonidos,





empezando con simples efectos sonoros que pueden incorporarse dentro de cualquier programa de juegos.

Se pueden utilizar una, dos o tres voces para conseguir estos efectos y enseguida veremos la forma en que pueden ser combinados. En un próximo capítulo de la sección de programación BASIC, describiremos detalladamente la forma de programar eficazmente el *chip* **AY-3-8910**.

Este *chip* es lo bastante potente como para permitir la creación de una gran variedad de efectos sonoros interesantes con una cantidad de programación relativamente pequeña. En lo que sigue vamos a centrarnos en el

manejo directo del *chip* a través de sus registros y del uso de la instrucción SOUND. Esta instrucción, cuya sintaxis es:

```
SOUND n.registro,valor
```

sirve para colocar determinados valores en los distintos registros del *chip*. Su utilización es a veces necesaria pues permite llegar a donde no llega la instrucción PLAY. Por otro lado resulta muy interesante ver las grandes diferencias de sonido que se pueden conseguir alterando, aunque sólo sea ligeramente, los contenidos de algunos de los registros. Te recomendamos que para trabajar con los registros tengas delante una tabla de los mismos, como la que publicamos en la página 11 del número uno de **INPUT**.

Para cada sonido se requieren en general hasta cinco o seis asignaciones a determinados registros. Para valorar los efectos de cada una teclea el siguiente programa:

```
100 SOUND6,30
105 SOUND8,16
110 SOUND7,&B10110111
112 SOUND12,45:SOUND11,0
115 SOUND 13,14
120 FOR J=1 TO 5000:NEXT
130 GOTO 100
```

Ejecuta el programa (RUN) y podrás oir el inconfundible rumor de las olas rompiendo contra el acantilado. Vamos a ver cómo actúa cada registro. La primera asignación, al registro 6, determina el periodo de la señal de ruido, que es la que da al sonido que escuchas ese característico timbre de «ola de mar». El valor asignado a este registro (que puede ser un número de 0 a 63) es en este caso 30. Podemos decir que es responsable del tamaño de las olas. Prueba a introducir un valor más pequeño y verás como parece que rompen olas más pequeñas. Y lo mismo, prueba a introducir un valor más grande y escucharás el rumor de olas de tremendas dimensiones.

El siguiente registro, el 8, controla la mezcla de la señal del canal A con el generador de envolventes. La siguiente asignación es el registro 7. Este controla la asignación de ruido o tono a cada uno de los 3 canales de sonido A, B y C. En este caso, y sin considerar los 2 bits de la izquierda que tienen otros cometidos, el 0 del cuarto bit determina que por el canal A se escuche la señal de ruido.

Los registros 11 y 12 determinan el periodo de repetición de la envolvente. En este caso, esta repetición determina algo así como la cadencia de las olas. Introduce en 12 un valor mayor y verás cómo la frecuencia de llegada de olas disminuye. Por último el registro 13 fija la forma de la envolvente. En nuestro ejemplo hemos escogido una envolvente triangular.



Vamos a ver cómo varía el sonido con unas ligeras modificaciones en los registros. Teclea lo que sigue:

```
100 SOUND6,30
105 SOUND8,16
110 SOUND7,&B10110111
112 SOUND12,4:SOUND11,0
115 SOUND 13,14
120 FOR J=1 TO100:NEXT
130 GOTO 100
```



Escribe RUN y escucha. Las olas han dado paso a un tren a vapor. Fíjate que hemos utilizado los mismos registros, y que en algunos de ellos no hemos cambiado el valor. Lo que sí tienes que tener en cuenta es que, además de los registros estamos utilizando el bucle de retardo de la línea 120. Prueba a hacer experimentos. Cambia el valor del registro 12 y verás cómo cambia la velocidad del tren.

SONIDOS PARA EL LABERINTO

Si tienes almacenado el juego de laberintos que vimos en el capítulo anterior, puedes añadirle ahora las siguientes líneas después de cargar (LOAD) el programa.

```
10 SOUND 1,0
12 SOUND 8,0
14 SOUND 7,&B11111110
2000 VI=VI-1:SOUND 8,15
     :FOR Z=1 TO 100:VPOKE
     HA,Z:SOUND 0,2*Z:FOR
     ZZ=1 TO 10:NEXT:NEXT
     :VPOKE HA,32:SOUND8,0
3005 SOUND 8,16:SOUND 13,1
     :SOUND 0,60:SOUND 1,0
     :SOUND 11,230:SOUND
     12,35:
```

Las tres primeras líneas se encargan de inicializar el *chip* de sonido al comienzo del juego. La nueva línea 2000, que sustituye a la línea 2000 del programa del capítulo anterior, incluye la generación de un tono, por el canal A, cuyo periodo va aumentando (el sonido se va haciendo más grave) a medida que se POKEan valores en el registro 0. Por su parte, la línea 3005, crea una especie de ¡PING! cada vez que el buscador de tesoros llega a su objetivo antes de que se cumpla el tiempo límite.

Prueba estos sonidos y di lo que te parecen. Verás como, aún tratándose de sonidos muy sencillos y fáciles de programar, le dan mucha gracia al juego. He aquí otros efectos de sonido con los que te gustará experimentar y que podrás incorporar a cualquiera de tus juegos. El primero es el sondio de un helicóptero.

```
5 I=2.5
10 SOUND 7,&B00110111
20 SOUND 8,14
30 FOR L=63 TO 1 STEP -I
   :SOUND 6,L:NEXT L
```

```
35 A$=INKEY$:IF A$="q"
   THEN I=I+.1:IF I>6.5
   THEN I=6.5
36 IF A$="z" THEN I=I-.1
   :IF I<1.5 THENI=1.5
40 GOTO 10
```

El sonido del helicóptero se consigue mediante el generador de ruido, que ha sido asignado al canal A. Desde el bucle de la línea 30 se hace variar rápidamente el periodo de la señal de ruido (actuando sobre el registro 6) lo que produce un sonido muy similar al de un helicóptero. Las líneas 35 y 36 te permitirán, actuando sobre las teclas **q** y **z**, acelerar o decelerar el helicóptero. Esto te puede dar una idea de cómo modificar los parámetros de un sonido desde el interior de un programa. ¿Imaginas lo bien que quedaría este sonido si sobre la pantalla apareciera la imagen un helicóptero? Podrías hacer que al mismo tiempo que el sonido se acelera, tu helicóptero iniciara el ascenso hacia la parte superior de la pantalla.

Vamos con otro ejemplo.

```
10 N=240
20 SOUND 7,&B00111110
30 SOUND 8,14
40 SOUND 0,170:SOUND 1,0
50 FOR J=1 TO N:NEXT
60 SOUND 0,150:SOUND 1,0
70 FOR J=1 TO N:NEXT
80 A$=INKEY$:IF A$="q"
   THEN N=N+30
90 IF A$="z" THEN N=N-30
   :IF N<1 THEN N=1
100 GOTO 40
```

Al escribir RUN podrás escuchar el sonido de la sirena de un coche de la policía. La rutina utiliza en este caso uno de los generadores de tono, el del canal A. El generador de ruido no interviene. En la línea 40 se fija un periodo para el tono, el que corresponde al ciclo grave de la sirena. Durante el tiempo que dura el bucle de la línea 50 suena el tono grave. A continuación, en la línea 60, se fija un nuevo periodo, esta vez el del ciclo agudo. Al alternar estos dos ciclos, grave y agudo, se produce una buena imitación del sonido de la sirena. Por último, las líneas 80 y 90 se ocupan de modificar la duración de cada uno de los ciclos cuando se pulsan las teclas **q** y **z**. Prueba al pulsarlas y verás como cambia la duración de cada ciclo de la sirena.

El ejemplo que te vamos a presentar ahora, está indicado para los juegos en los que se utiliza un arma laser. Teclea lo que sigue:

```
5 FOR J=150 TO120 STEP -3
10 SOUND0,J:SOUND 1,1
20 SOUND6,10
30 SOUND7,&B00110110
40 SOUND 8,16
50 SOUND 11,100:SOUND 12,8
60 SOUND 13,4
70 FOR K=1 TO 170:NEXT
80 NEXT
```

En este caso se escucha una mezcla, a través del canal A, de las señales de uno de los generadores de tono y del generador de ruido blanco. La mezcla de estas dos señales se modela en volumen mediante una envolvente en rampa (esta envolvente es la número 4 de las predefinidas en el *chip*). La envolvente es la responsable de que la ráfaga de la laser vaya aumentando paulatinamente de volumen, hasta llegar a su volumen máximo, momento en el que el sonido se corta bruscamente. Por último, el bucle FOR...NEXT, de las líneas 5 y 80, determina que el periodo del tono vaya disminuyendo a cada ráfaga. De esta forma el sonido del laser resulta mucho más sideral e intergaláctico.

Por último, te ofrecemos una rutina de sonido en la que se utilizan valores aleatorios para el periodo de las señales de los tres generadores de tono.

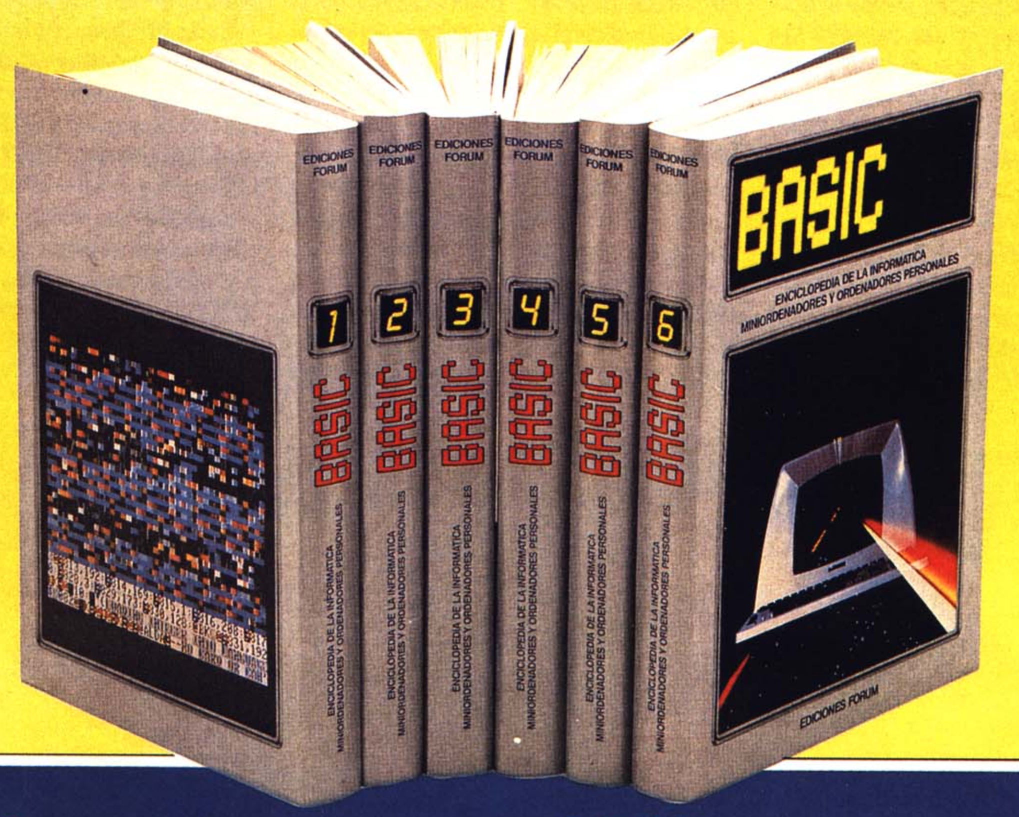
```
10 SOUND 7,&B00111000
20 SOUND 8,15:SOUND 9,15
   :SOUND 10,15
30 R=RND(-TIME)
40 F1=10+(90*RND(1))
   :F2=50+(80*RND(1))
   :F3=180+(50*RND(1))
50 SOUND 0,F1:SOUND 2,F2
   :SOUND 4,F3
60 FOR J=1 TO 40:NEXT
70 GOTO 40
```







ENCICLOPEDIA DE LA INFORMATICA DE LOS MINIORDENADORES Y ORDENADORES PERSONALES



84 fascículos semanales de 24 páginas cada uno.

6 volúmenes de gran formato (19,5 × 27,5) encuadernados en geltex impreso a todo color.

1.748 páginas en papel especial.

2.000 gráficos e ilustraciones a color.

BASIC

Una información indispensable del primero al último fascículo.

BASIC

Para no ser un extraño en el futuro tecnológico que nos aguarda.

BASIC

Para poner una nueva ciencia a nuestro servicio.

SUPER OFERTA DE LANZAMIENTO

Con su primer fascículo recibirá GRATIS el fascículo n.º 2, es decir, su primer envío constará de 5 fascículos al precio de 4.

RECORTE ESTE CUPON Y ENVIELO A EDISA (Dpto. de Suscripciones), López de Hoyos, 141 - 28002 Madrid

SI, deseo suscribirme a **BASIC** y recibiré en mi hogar 4 fascículos al mes, abonando sólo **700 Ptas.** por cada envío. El servicio a mi domicilio es GRATUITO.

▽ POR FAVOR, RELLENE SUS DATOS EN MAYUSCULAS ▽

NOMBRE ______

APELLIDOS ______

DIRECCION ______ N.º ______

PISO ______ COD. POSTAL ______

CIUDAD ______

PROVINCIA ______

TLFNO. ______

FIRMA ______

ORDENACION, INDEXACION, BUSQUEDA E INSERCION

Si estás preparando, o tienes pensado hacerlo, algún programa de archivo, te será de gran ayuda conocer las distintas técnicas de ordenación, búsqueda e inserción de datos. Conociéndolas podrás aplicarlas, en cada caso, de la forma más adecuada, ahorrando tiempo y evitándote problemas.

Cuando se maneja una cantidad reducida de información podemos almacenarla, consultarla o alterarla sin grandes problemas, sin preocuparnos mucho de utilizar unos procedimientos excesivamente rígidos. Puede bastarnos un lapicero y un papel para anotar nuestras necesidades de compra en unos grandes almacenes. Todo nuestro problema se resolverá anotando las cosas a medida que se nos ocurren y consultando la lista cada vez que pasemos por una sección donde vendan algo que recordemos vagamente que tenemos que comprar.

Por lo general no tendremos dificultades especiales aunque en nuestra lista tengamos la siguiente relación: bañador Luis, tazones desayuno, melones, harina, toallas pequeñas, dos paquetes de café molido.

Si escribiésemos con la misma alegría la lista de las personas y teléfonos de una ciudad seguramente nos veríamos en aprietos a la hora de localizar a cualquiera de ellas. La compania Telefónica por lo menos así lo entiende y por ello nos ofrece dicha lista ordenada por apellidos, calles o profesiones (para el uso de su propio personal utiliza incluso otra ordenada por números de teléfono).



Para que una información de cierto volumen sea útil, conviene que esté «ordenada» por algún criterio. Si no está ordenada puede servir, pero a costa de que el procedimiento de búsqueda sea más lento. Si las variaciones de una lista son numerosas y frecuentes, y el procedimiento de ordenación laborioso puede que no nos compense la ordenación si las consultas van a ser totalmente esporádicas. No es infrecuente que en el cuaderno que tenemos al lado del teléfono de casa vayamos apuntando nombres y teléfonos uno a continuación de otro, y que, cuando ya no nos sirven, los eliminemos con una simple tachadura. Este procedimiento nos obliga a coger la lista desde el principio y «pasear el dedo» hasta encontrar lo que buscamos sin que haya gran diferencia con utilizar una agenda de clasificación alfabética. A la hora de añadir nombres, si la lista no es muy grande, pueden colocarse al final o «entre» renglones ya existentes. El procedimiento funciona.

Cuando se trata de localizar información sobre las diferentes partes de un libro existe también la posibilidad de «hojearlo» hasta encontrar la información que nos interesa, aún cuando este procedimiento no es aconsejable si el libro es de un cierto tamaño o si las consultas son frecuentes. La solución adecuada es disponer de un «índice» donde encontraremos clasificados por orden alfabético o temático, los aspectos que nos interesen.

Esta necesidad de «ordenación», «indexación», «búsqueda» e «inclusión» se pone de manifiesto a la hora del manejo de bloques de información (ficheros) por los ordenadores. Todas estas operaciones pueden hacerse directamente en memoria central o incluso en el propio *diskette* (o disco duro). En éste úlitmo caso podremos obviar las limitaciones de memoria pero a costa de un alto precio en el desgaste del material y sobre todo en los tiempos de respuesta.

El manejo de datos en memoria central puede hacerse fundamentalmente con ayuda de variables subindicadas (numéricas o de caracteres), variables simples de caracteres y **disco**



RAM. En este artículo nos vamos a ceñir a las primeras por ser las de uso más generalizado.

Cuando hablamos de variables subindicadas nos referimos a vectores (listas), matrices de dos dimensiones (tablas) y a estructuras de más de tres dimensiones, si bien estas últimas por lo general se descomponen por comodidad en matrices bidimensionales.

BUSQUEDA E INCLUSION

La búsqueda en una lista es **secuencial** cuando para encontrar un determinado nombre (o valor numérico) se empieza por la primera posición y se van analizando todas hasta que se encuentra lo que buscamos, o hasta el final de la lista en el caso de que pueda haber más elementos que cumplan la condición impuesta (por ejemplo, todos los nombres que comiencen por MART). Este procedimiento es sólo adecuado cuando la lista está desordenada.

El mecanismo de búsqueda en una matriz A$( ) sería:

```
10 FOR I=1 TO N
20 IF A$(I)="Nombre buscado"
   THEN PRINT I
30 NEXT I
```

Los sucesivos valores de **i** nos indicarían los lugares de aparición del nombre buscado dentro de la lista.

Si disponemos de una lista ordenada podemos hacer una búsqueda más rápida mediante un sistema **dicotómico**, **binario** o de **corte**. Se toma el valor central de la lista y se compara con el que estamos buscando. Si éste es menor descartamos la segunda mitad de la lista y repetimos el proceso dividiendo por la mitad hasta que nos quedemos en un solo valor. Si este último valor es igual al que buscamos habrémos hallado en qué lugar empieza en la lista y podremos hacer, a partir de este punto una búsqueda secuencial de valores iguales hasta que aparezca un valor distinto. También puede ocurrir que el nombre que buscamos no exista, pero al menos sabremos donde debería aparecer de haber estado.

Una rutina para esta búsqueda puede ser:

```
1 N=10:REM ELEGIR NUMERO DE
  ELEMENTOS A EXPLORAR
10 P=1:U=N:R=U
20 R=INT((U-P)/2):M=P+R
30 IF X$<=A$(M) THEN P=P:U=M
   ELSE P=M+1:U=U
40 IF P<U THEN 20
50 IF X$=A$(M) THEN PRINT X$
   +" APARECE EN ";M ELSE
   PRINT X$+" NO EXISTE EN LA
   LISTA"
```

Este procedimiento de búsqueda es realmente rápido. El número de pasadas depende de la longitud de la lista.

Es normal que en algún momento se presente la necesidad de eliminar o añadir algún elemento a la lista. Con el procedimiento de búsqueda binaria sabremos donde eliminar y proceder al correspondiente corrimiento para ocupar el espacio vacío e igualmente donde colocar el nuevo valor, haciendo el hueco correspondiente. La ventaja radica en que el nuevo valor que añadamos queda ya ordenado.

ORDENACION

Un ejemplo de conjunto desordenado sería un mazo de cartas bien barajado.

Una lista decimos que está ordenada cuando sus elementos se hallan dispuestos uno a continuación de otro siguiendo un criterio, como puede ser el alfabético. Los ordenadores utilizan por lo general como referencia el código ASCII donde los números tienen valores inferiores a las letras mayúsculas y éstas a las minúsculas (prácticamente todos los manuales de ordenadores disponen de una tabla ASCII).

Existen varios métodos o procedimientos de ordenación (*sorting* en la literatura inglesa). Cada uno de ellos tiene sus peculiaridades y una recomendación concreta. Unos son más fáciles de comprender y programar, otros son más rápidos a costa de utilizar más memoria RAM, otros por el contrario son parcos en consumo de memoria pero más lentos. A continuación comentaremos algunos de ellos de forma que podamos sacar unas ideas claras que nos permitan ayudarnos a programar nuestras necesidades o juzgar algunas rutinas existentes. Al final hemos incluido un programa donde de una manera muy gráfica podemos experimentar con la ordenación.

METODO DE SUSTITUCION

Supongamos una lista (vector o matriz de una dimensión) de **n** elementos.

El método consiste en ir comparando cada elemento con todos los que le siguen (1 con 2, 1 con 3, 1 con 4, etc.),



permutándolos si el segundo es menor que el primero. Al final de la pasada el primer elemento será el de menor valor. En pasadas sucesivas se repite lo mismo pero desde el segundo lugar (2 con 3, 2 con 4, 2 con 5, etc.) y así sucesivamente. Una vez terminado el ciclo, la lista quedará ordenada de menor a mayor.

La lista a ordenar se carga en la matriz A$( ) de dimensión **N**, y la rutina de ordenación queda así:

```
10 N=5:DIM A$(N)
20 FOR I=1 TO N:READ A$(I):
   NEXT I
30 FOR I=1 TO (N-1)
40   FOR H=I+1 TO N
50   IF A$(I)>A$(H) THEN SWAP
       A$(I),A$(H)
60   NEXT H
70 NEXT I
100 FOR I=1 TO N:PRINT A$(I)
     :NEXT I:STOP
110 DATA CF,AA,DE,SE,ZX
```

Este sería el aspecto de las diversas pasadas del proceso

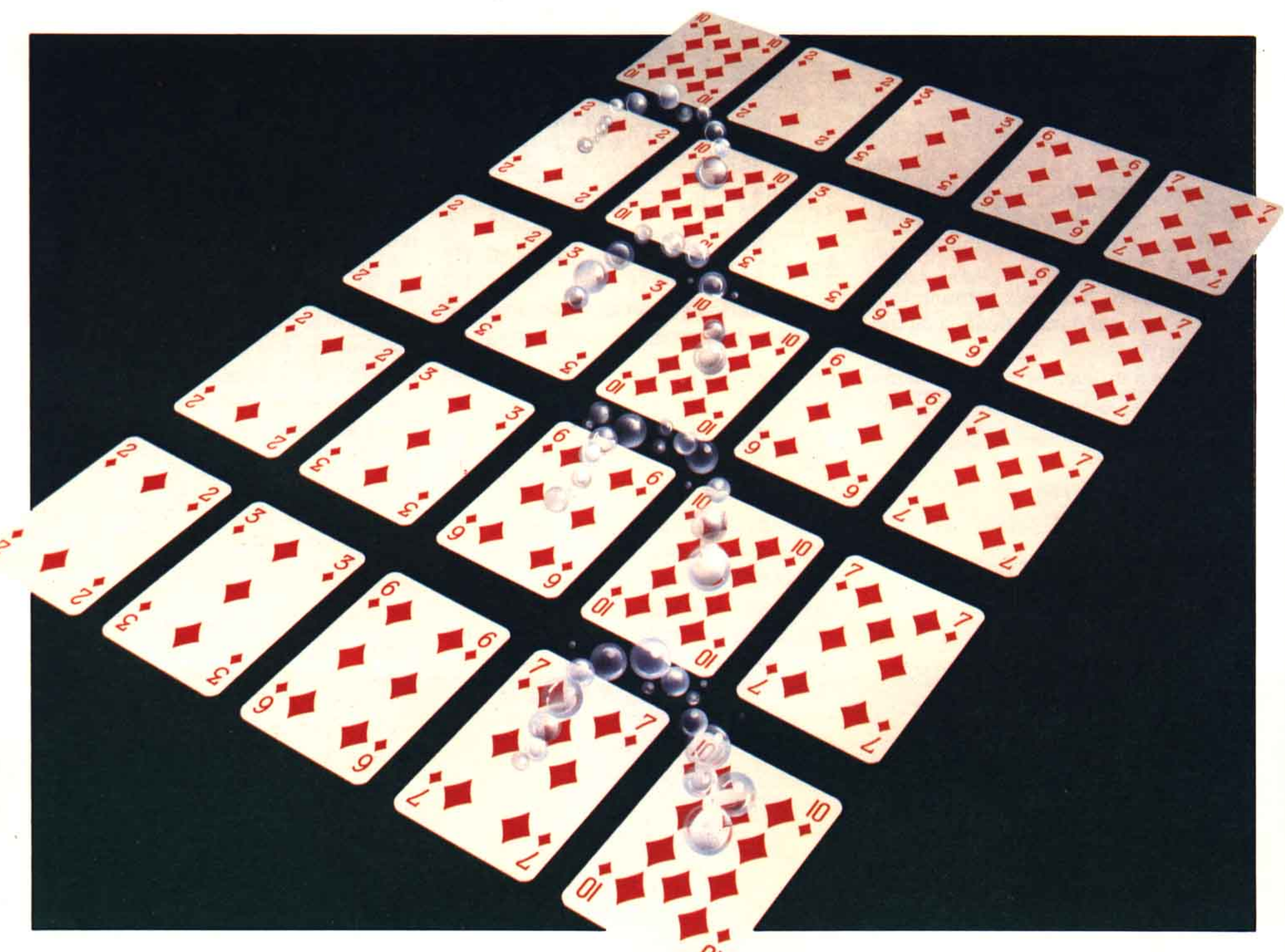
```
(CF)(AA)(AA)(AA)  AA   AA   AA
(AA) CF   CF   CF  (CF)(CF)(CF)
 DE (DE) DE   DE  (DE) DE   DE
 SE  SE (SE)  SE   SE (SE)  SE
 ZX  ZX  ZX  (ZX)  ZX   ZX  (ZX)
 AA  AA    AA     AA
 CF  CF    CF     CF
(DE)(DE)   DE     DE
(SE) SE   (SE)    SE
 ZX (ZX)  (ZX)    ZX
```

Este método es el más elemental e intuitivo pero, sin lugar a dudas el más lento. Cuando rodemos el programa de demostración del final veremos que para un número reducido de elementos puede ser suficientemente rápido. En este programa aparece el proceso a velocidad normal (muy rápido), ralentizado por el hecho de aparecer en pantalla los diversos pasos (rápido) y ralentizado a propósito para ver a **cámara lenta** el proceso (lento).

METODO DE LA BURBUJA

Quizá es uno de los métodos más populares. Consiste en ir comparando parejas sucesivas de valores (1y2, 2y3, 3y4 ... 9y10) e intercambiarlos si el





primero es mayor que el segundo. Al final de la primera pasada el último valor será el mayor de la lista. La segunda pasada se hará por lo tanto para un valor menos y así sucesivamente. Si la ordenación se hace de mayor a menor, las pasadas sucesivas harán ascender los números menores como si se tratase de las burbujas de un líquido; de ahí su nombre.

Si al final de una de las pasadas no se ha producido la necesidad de intercambio de valores quiere decir que la lista entera está ordenada y se puede suspender el proceso con el consiguiente ahorro de tiempo. Esta es una de las ventajas principales. Su uso está indicado por tanto en aquellos casos en que la lista es grande pero está casi ordenada (por ejemplo al añadir un elemento a una lista ordenada).

```
10 N=5:DIM A$(N)
20 FOR I=1 TO N:READ A$(I):
   NEXT I
30 FOR I=1 TO (N-1):K=0
40    FOR H=1 TO (N-I)
50    IF A$(H)>A$(H+1) THEN
      SWAP A$(H),A$(H+1):
      K=K+1
60    NEXT H:IF K=0 THEN
      I=N-1
70 NEXT I
100 FOR I=1 TO N:PRINT A$(I):
    NEXT I:STOP
110 DATA CF,AA,DE,SE,ZX
```

Este sería el aspecto de las diversas pasadas del proceso

```
(CF) AA   AA   AA
(AA)(CF)  CF   CF
 DE (DE)(DE)  DE
 SE   SE (SE)(SE)
 ZX   ZX   ZX (ZX)

(AA) AA   AA   AA
(CF)(CF)  CF   CF
 DE (DE)(DE)  DE
 SE   SE (SE)(SE)
 ZX   ZX   ZX (ZX)
```

METODO «SHELL»

Existen variantes de estos métodos como puede ser el método **Shell** en donde el aumento de velocidad es considerable. En este caso se toma como referencia la mitad de la lista y se van haciendo comparaciones entre parejas separadas entre sí el valor de referencia (el primero con el mitad, el segundo con el mitad más uno, etc.). Al final de la primera pasada la lista habrá quedado ordenada de una manera alternada. Variando la referencia al valor entero de la mitad de la referencia anterior, de forma sucesiva, se llega a una ordenación bastante más rápida, aunque realmente no se puede apreciar su justo valor más que en código máquina. En BASIC la lentitud de los bucles enmascara la mejora en velocidad.

Prueba con este programa:

```
10 N=10:D=N:DIM A$(N)
```

```
20 FOR I=1 TO 10:READ A$(I)
   :NEXT I
30 IF D<=1 THEN 100
40 D=INT(D/2):F=N-D
50 Z=0
60 FOR H=1 TO F
70 IF A$(H)>A$(H+D)
   THEN SWAP A$(H),A$(H+D)
   :Z=1
80 NEXT H
90 IF Z>0 THEN 50 ELSE
   30
100 FOR I=1 TO N:PRINT A$(I)
    :NEXT I:STOP
110 DATA SERT,DERT,CFGT,ZSDL,
    AASER,XSD,DE,SERTY,FF,
    AZZW
```

METODO «SHELL-METZNER»

Se trata de una variante del método «Shell» con la que se consigue una mayor velocidad.

Para el mismo caso anterior:

```
10 N=10:DIM A$(N)
20 FOR I=1 TO N:READ A$(I):
   NEXT I
30 D=INT(N/2)
40 IF D<1 THEN 100
50 F=N-D
60 FOR I=1 TO F
70   FOR H=I TO 1 STEP -D
80   IF A$(H+D)<A$(H) THEN
     SWAP A$(H),A$(H+D)
90   NEXT H:NEXT I
95 D=INT(D/2):GOTO40
100 FOR I=1 TO N:PRINT A$(I)
    :NEXT I:STOP
110 DATA SERT,DERT,CFGT,ZSDL,
    AASER,XSD,DE,SERTY,FF,
    AZZW
```

Como curiosidad, sobre todo si no disponemos de ordenador, podemos experimentar los métodos anteriores con ayuda de las cartas de una baraja. Elegimos, por ejemplo, cinco cartas cualesquiera y las colocamos vertical u horizontalmente. Luego las vamos moviendo e intercambiando de acuerdo con las indicaciones de la rutina correspondiente.

ORDENACION RAPIDA

Hasta ahora hemos considerado sólo la posibilidad de reordenar la propia matriz, sin ningún gasto adicional de memoria.

Si nos encontramos con un cajón lleno de fichas que hemos de clasificar, como método alternativo a la reclasificación en el propio cajón podemos seguir el procedimiento de tomar otro cajón vacío e ir pasando fichas del primero al segundo pero dejándolas ya cada una en su lugar. El proceso es evidentemente más rápido y no hay que dar tantas vueltas a las fichas.

Con un ordenador el proceso es análogo. Necesitamos crear otra matriz auxiliar de capacidad adecuada (si la lista es grande podemos tener problemas de memoria) a donde iremos pasando una a una las posiciones de la primera matriz, ayudándonos de una búsqueda binaria para hacer la inclusión en el lugar oportuno.

INDEXACION

En el caso de listas con varios campos en cada registro o tablas (matrices de dos dimensiones) puede surgirnos la necesidad de ordenarlas según diversos criterios (apellidos, profesiones, calles, número de cliente, etc.). Bastaría con hacer tantos procesos de ordenación como fuera preciso tal como se ha visto en el apartado anterior. El único problema sería el elevado volumen de información redundante que tendríamos que manejar.

Lo normal en este caso es conservar la lista (matriz) inicial y confeccionar tantos índices como sea preciso. Cada índice estará compuesto por una referencia inequívoca (no valdría un apellido ya que puede haber más de un cliente con el mismo; si valdría el D.N.I. o el número de cliente) y el contenido del campo respectivo. Por ejemplo el número de referencia de cliente y los apellidos, o el número de referencia de cliente y el número de teléfono, o la ciudad etc. Uno de los índices será también el número de cliente asociado al número de orden que ocupe ese cliente en la tabla.

Cuando deseemos saber el número de teléfono de un cliente en particular iremos primero a la matriz de índice-apellido y buscarémos que número de cliente le corresponde. A continuación consultaremos el índice-número de cliente el cual nos informará donde encontrar la ficha completa de dicho cliente, incluido su teléfono.

Esta forma de proceder, aunque un poco laboriosa, es la única que nos permite manejar un gran volumen de información con un consumo razonable de memoria.

Las Bases de datos hacen uso de estos principios manejando informaciones almacenadas en ficheros residentes en *diskettes* o discos duros con los correspondientes pasos por memoria central.

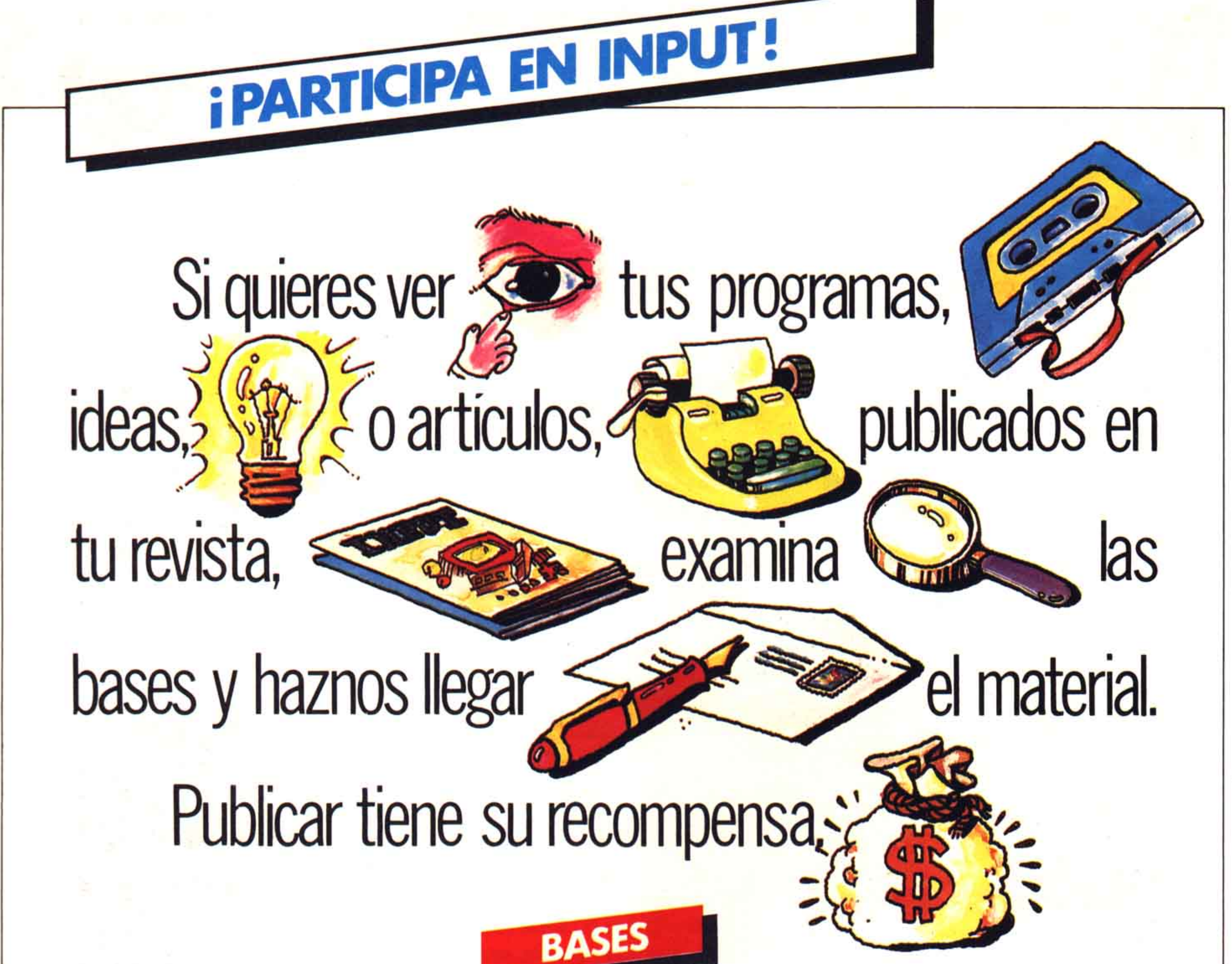
Lo normal es que los índices se vayan actualizando automáticamente a medida que se añade o elimina información de la base de datos.

EL PROGRAMA

Como hemos comentado, incluímos un programa que muestra el funcionamiento de los distintos algoritmos de ordenación. El objetivo del mismo es, sobre todo, ser didáctico. En la pantalla se muestra paso a paso el proceso de ordenación de una tabla de 10 elementos. Tecléalo, experimenta y saca tus propias conclusiones sobre las ventajas de utilizar uno u otro método.

```
20 CLS:SCREEN 0:COLOR1,11,11
   :WIDTH 32:KEY OFF
30 DIM A$(10),B$(10)
100 'FORMACION DE LA LISTA
110 FOR I=1 TO 10
120 LOCATE 0,17:PRINT
    "INTRODUCE DIEZ NOMBRES"
130 GOSUB 800:INPUT"NOMBRE
    (max.5c)";A$(I):IF LEN
    (A$(I))>5 THEN A$(I)=
    LEFT$(A$(I),5)
140 B$(I)=A$(I):LOCATE 0,I+2
    :PRINT I TAB(4) A$(I)
150 NEXT I
160 CLS: PRINT "OPCIONES
    METODOS ORDENACION :"
```

¡PARTICIPA EN INPUT!



BASES

PROGRAMAS: Una vez desarrollado tu programa, que debe ser original y no haber sido enviado a ninguna otra publicación, puedes enviárnoslo aquí grabado en cassette, diskette o microdrive. Es preferible que vaya acompañado por un listado de impresora, pero no es imprescindible.

El programa habrá de venir acompañado por un texto que aclare cuál es su objetivo, el modo de funcionamiento y una explicación del cometido que cumplen las distintas rutinas que lo componen. El texto se presentará en papel de tamaño folio y mecanografiado a dos espacios. No importa que la redacción no sea muy clara y cuidada; nuestro equipo de expertos se encargará de proporcionarle la forma más atractiva posible.

ARTICULOS E IDEAS: Se aplica lo anteriormente dicho para los textos que acompañan a los programas; es decir, conviene detallar al máximo lo que desees que aparezca publicado en la revista, de la manera que te gustaría que otra persona hubiera explicado eso mismo.

UN JURADO propio decidirá en cada momento qué colaboraciones reúnen los requisitos adecuados para su publicación, y evaluará la cuantía del premio en metálico al que se hagan acreedoras.

No olvidéis indicar claramente para qué ordenador está preparado el material, así como vuestro nombre y dirección y, cuando sea posible, un teléfono de contacto. Entre todos los trabajos recibidos durante cada mes SORTEAREMOS:

- **Un premio de 50.000 ptas.**
- **Un premio de 25.000 ptas.**
- **Un premio de 10.000 ptas.**
  en material microinformático a elegir por los afortunados.

¡No os desaniméis!, por muy simples o complejas que puedan parecer vuestras ideas, todas serán revisadas con el máximo interés.

INPUT MSX
Alberto Alcocer, 46, 4.º B
28016 Madrid

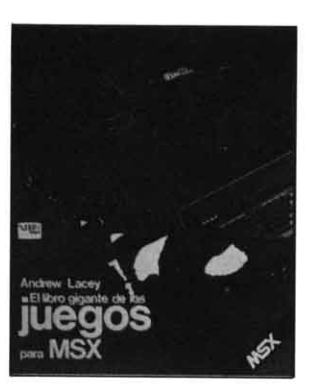
NOTA: INPUT no se responsabiliza de la devolución del material que no vaya acompañado por un sobre adecuado con el franqueo correspondiente.

Ordena tus propias ideas

Le sacarás partido a tu ordenador



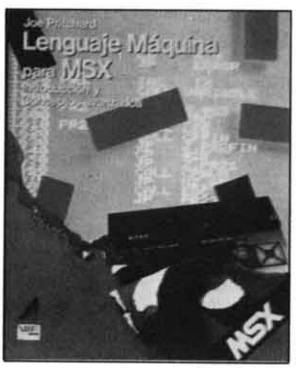
DESCUBRE TU MSX.
Programación y aplicaciones
Joe Pritchard
1.272 ptas.



EL LIBRO GIGANTE DE LOS JUEGOS PARA MSX.
Andrew Lacey
1.590 ptas.



MSX: GUIA DEL PROGRAMADOR Y MANUAL DE REFERENCIA.
T. Sato, P. Mapstone e I. Muriel
2.279 ptas.



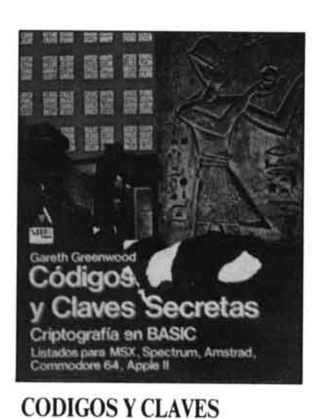
LENGUAJE MAQUINA MSX. INTRODUCCION Y CONCEPTOS AVANZADOS.
Joe Pritchard
1.537 ptas.



EL SUPERLIBRO DE LOS JUEGOS PARA ORDENADOR
Tim Hartnell
2.120 ptas.



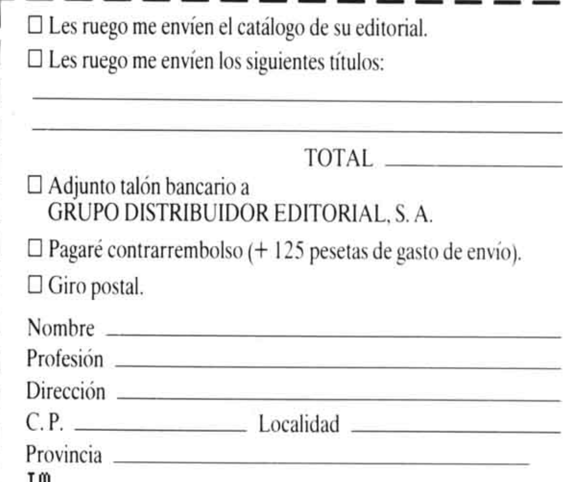
SIMULACIONES: REPLICA LA REALIDAD CON TU ORDENADOR
Tim Hartnell
1.643 ptas.



CODIGOS Y CLAVES SECRETAS: CRIPTOGRAFIA EN BASIC.
Listados para MSX, Spectrum, Amstrad, Commodore 64, Apple II
Gareth Greenwood
1.378 ptas.



COMETAS EN TU MICRO: EL HALLEY.
Cálculos de órbitas y parámetros de cometas en BASIC.
F. Galende, A. Sánchez, M. Alparaz y J. A. Sánchez García
550 ptas.



☐ Les ruego me envíen el catálogo de su editorial.
☐ Les ruego me envíen los siguientes títulos:
____
____
TOTAL __________
☐ Adjunto talón bancario a GRUPO DISTRIBUIDOR EDITORIAL, S. A.
☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).
☐ Giro postal.
Nombre __________
Profesión __________
Dirección __________
C.P. __________ Localidad __________
Provincia __________
IM



Adquiéralos en su librería habitual.
Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a:
Apdo. de Correos 14632, Ref. D. de C. 28080 MADRID

ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA

```
        :PRINT:PRINT
170 PRINT"1 SUSTITUCION
    (LENTO)"
180 PRINT"2 SUSTITUCION
    (RAPIDO)"
190 PRINT"3 SUSTITUCION
    (MUY RAPIDO)"
200 PRINT"4 BURBUJA.....
    (LENTO)"
210 PRINT"5 BURBUJA.....
    (RAPIDO)"
220 PRINT"6 BURBUJA.....
    (MUY RAPIDO)"
230 PRINT:PRINT:PRINT"PULSE
    OPCION"
240 P$=INKEY$:IF P$="" OR P$>
    "6" THEN 240
250 P=VAL(P$):ON P GOSUB 300,
    300,350,400,400,500
260 GOTO 160
300 CLS:GOSUB 2000:GOSUB 2020
    :GOSUB 820
310 FOR N=1 TO 9:FOR H=N+1 TO
    10
315 GOSUB 1000:GOSUB 2020:
    GOSUB 760
320 IF A$(N)<=A$(H) THEN
    GOSUB 990:GOTO  330
325 IF P=1 THEN GOSUB 700:
    SWAP A$(H),A$(N)
326 IF P=2 THEN SWAP A$(H),
    A$(N)
330 LOCATE 12,H:PRINT SPACE$
    (5):LOCATE 12,N:PRINT
    SPACE$(5)
335 GOSUB 990:GOSUB 2020:
    GOSUB 1000
340 NEXT H:NEXT N:GOSUB 2020:
    GOSUB 830:GOSUB 800:PRINT
    "CUALQUIER TECLA PARA
    SEGUIR"
345 IF INKEY$="" THEN 345
    ELSE RETURN
350 CLS:GOSUB 2000:GOSUB 820
360 FOR N=1 TO 9:FOR H=N+1 TO
    10
370 IF A$(N)>A$(H) THEN SWAP
    A$(H),A$(N)
380 NEXT H:NEXT N:GOSUB 2020:
    GOSUB 830:GOSUB 800:PRINT
    "CUALQUIER TECLA PARA
    SEGUIR"
390 IF INKEY$="" THEN 390
    ELSE RETURN
400 CLS:GOSUB 2000:GOSUB 2020
    :GOSUB 820:R=9
410 S=0:FOR N=1 TO R: H=N+1
415 GOSUB 1000:GOSUB 2020:
    GOSUB 760
420 IF A$(N)<=A$(H) THEN
    GOSUB 990:GOTO  430
425 IF P=4 THEN GOSUB 700:
    SWAP A$(H),A$(N):S=S+1
426 IF P=5 THEN SWAP A$(H),A$
    (N):S=S+1
430 LOCATE 12,H:PRINT SPACE$
    (5):LOCATE 12,N:PRINT
    SPACE$(5)
435 GOSUB 990:GOSUB 2020:
    GOSUB 1000
440 NEXT N:R=R-1:IF R=0 OR S<
    >0 THEN 410 ELSE GOSUB
    2020:GOSUB 830: GOSUB 800
    :PRINT"CUALQUIER TECLA
    PARA SEGUIR"
445 IF INKEY$="" THEN 445
    ELSE RETURN
500 CLS:GOSUB 2000:GOSUB 820:
    R=9
510 S=0:FOR N=1 TO R: H=N+1
520 IF A$(N)>A$(H) THEN SWAP
    A$(H),A$(N):S=S+1
540 NEXT N:R=R-1:IF R=0 OR S<
    >0 THEN 510 ELSE GOSUB
    2020:GOSUB 830: GOSUB 800
    :PRINT"CUALQUIER TECLA
    PARA SEGUIR"
545 IF INKEY$="" THEN 545
    ELSE RETURN
700 LOCATE 18,H:PRINT A$(H):
    GOSUB 990
710 LOCATE 18,H:PRINT SPACE$
    (5) TAB(25) A$(H):GOSUB
    990
720 LOCATE 18,H:PRINT A$(N)
    TAB(25) SPACE$(5)
730 LOCATE 18,N:PRINT SPACE$
    (5) TAB(25) A$(H)
740 GOSUB 990:LOCATE 18,N:
    PRINT A$(H) TAB(25)
    SPACE$(5)
750 RETURN
760 LOCATE 12,N:PRINT"---->
    " A$(N):LOCATE 12,H:PRINT
    "----> " A$(H) :RETURN
800 LOCATE 0,19:PRINT SPACE$
    (64):LOCATE 0,19:RETURN
820 LOCATE 3,12:PRINT"L.ORIG
    PROC.DE ORDENACION":
    RETURN
830 LOCATE 3,12:PRINT"L.ORIG
    L.ORD.";SPACE$(15)
    :RETURN
990 'RETARDO 50
995 IF P=1 OR P=4 THEN TIME=0
    ELSE TIME=50
997 IF TIME>50 THEN RETURN
    ELSE 997
1000 'BORRADO PROCESO
     ORDENACION
1010 FOR I=1 TO 10:LOCATE 18,
     I:PRINT SPACE$(13)
     :NEXT I:RETURN
2000 'PRINT LISTA ORIGINAL
2010 FOR I=1 TO 10:LOCATE 0,I
     :A$(I)=B$(I):PRINT I TAB
     (4) B$(I):NEXT I:RETURN
2020 'PRINT LISTA A
     CLASIFICAR
2030 FOR I=1 TO 10:LOCATE 12,
     I:PRINT A$(I):NEXT I
     :RETURN
```

DISKETTES EN LOS MICROORDENADORES

El uso de *diskettes* en un ordenador doméstico ha dejado de ser un lujo, gracias al descenso de los precios, para convertirse en una interesante posibilidad. En este artículo te contamos algunas cosas sobre los distintos tipos de *diskettes* y sobre las ventajas que supone utilizarlos.

El grabador doméstico de *cassettes* y la impresora térmica de rollo pequeño han hecho el milagro de poner la informática al alcance de cualquiera que tenga mucha ilusión y no tanto dinero.

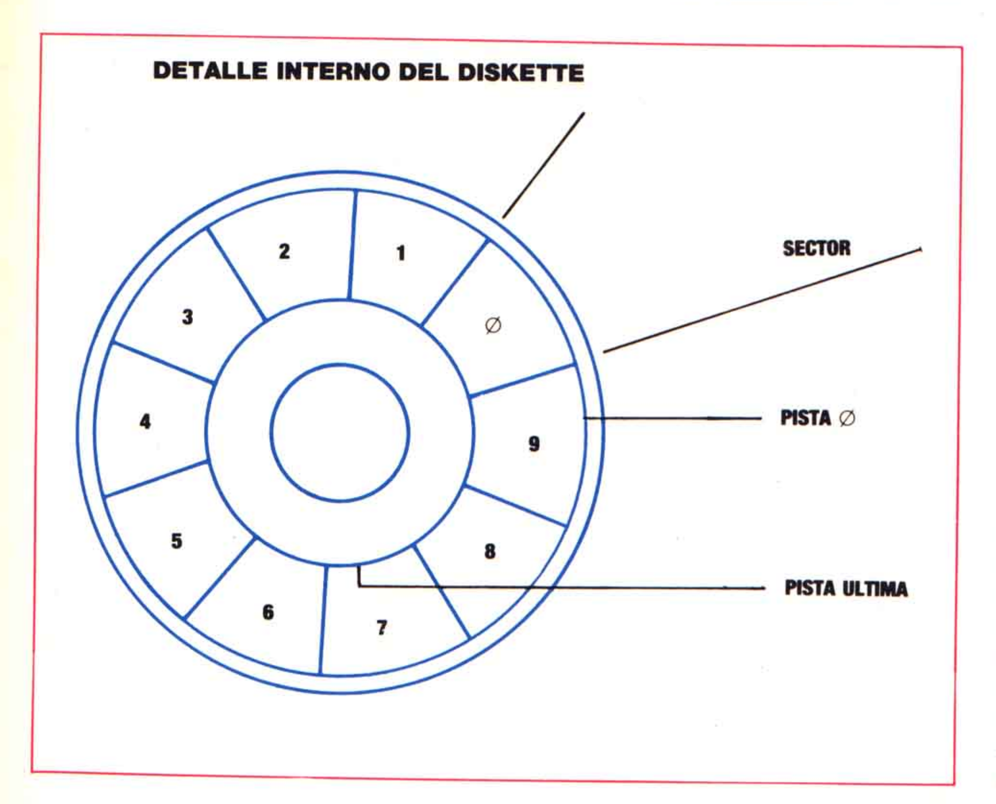
Pero cuando se quiere hacer algo realmente profesional, el *cassette* se queda corto, tanto en comodidad, rápidez, seguridad y presentación de resultados como en amplitud (trabajar sólo con las posibilidades de capacidad de la memoria RAM nos limita mucho el alcance de ficheros y la extensión de los programas).

Aunque no siempre es rigurosamente cierto, la mayor parte de las aplicaciones informáticas, y muy en especial las de gestión, requieren el manejo de una gran cantidad de datos y el uso de programas en los cuales se debe hacer uso frecuente de otros programas, subprogramas, rutinas, etc., que por su extensión, no puede residir permanentemente en memoria.

Lo cierto es que si queremos hacer trabajos profesionales necesitaremos herramientas profesionales y éstas pasan necesariamente por el *diskette* y la impresora (80 columnas y manejo de papel con perforaciones laterales de arrastre, como mínimo).

Nos asalta la terrible duda de si también necesitaremos adquirir un monitor profesional. La respuesta es simplemente no.

Un televisor doméstico de blanco y negro nos sobra y nos basta para la mayoría de las aplicaciones. No olvidemos que una gran parte de las pantallas de ordenadores profesionales son también bicolor (ambar/negro, verde/negro, etc.). Si el televisor está en condiciones normales, la definición es suficientemente buena como para manejar muy aceptablemente hasta 64 caracteres por línea. Recordemos que los estándares normales son sólo de 32 ó 40 caracteres por línea (80 en los profesionales de alta definición).



En el mercado español pueden encontrarse hoy día *diskettes* e impresoras profesionales (con sus correspondientes *interfaces*) a unos precios asequibles.

De todos los medios magnéticos para almacenamiento masivo de información (programas y datos) el *diskette* es el ideal desde un punto de vista de la relación precio/prestaciones.

Se trata de un dispositivo situado entre el *cassette* y el disco duro (o disco **Winchester**).

En esencia se trata de un soporte plástico flexible (por eso recibe también el nombre de disco flexible o *floppy disk*) recubierto de una película magnética análoga a la de los *cassettes*. Este soporte magnético tiene forma de disco y está diseñado para girar a una velocidad constante.

Los *diskettes* más utilizados son los de tamaño 5-1/4 pulgadas si bien empiezan a extenderse los de 3-1/2 pulgadas ya que aunque resultan más caros, ofrecen una mejor protección mecánica y sobre todo las unidades ocupan un espacio reducido, cosa aparentemente trivial pero muy importante en la práctica, ya que en la mesa de trabajo siempre nos falta espacio. Concretamente los discos de 3-1/2 son los utilizados en el estandar **MSX**, aunque modelos como el **Spectravideo** utilizan también de 5-1/4.

En cuanto a las capacidades también hay diferencias según se trate de uno u otro *diskette/drive*, pero en nuestro caso oscilarán entre 100 Kbytes y 400 Kbytes pudiendo manejarse hasta un máximo de 4 *drives*, lo cual nos da una gran capacidad de maniobra. Estos *drives* reciben generalmente los nombres de A, B, C, D. En **MSX** la capacidad del disco es de 360 Kbytes.

Para facilitar su manejo, el *diskette* viene provisto de una envoltura que le da rigidez moderada en los de 5-1/4 y rigidez elevada en los de 3-1/2 Esta envoltura sólo permite acceder al disco en tres zonas:

- Zona de lectura / escritura.

Es el canal por donde se deslizará la cabeza magnética que nos permitirá leer o grabar información. En los de 5-1/4 este canal permanece a la vista y en los de 3-1/2 queda protegido por una tapa que se abre automáticamente, sólo cuando se requiere el acceso a la información.

- Zona de sincronización.

Se trata de un orificio por donde se detecta, de una manera óptica, el paso de una o varias de las perforaciones practicadas en el *diskette*, lo que permite fijar con exactitud el comienzo de cada vuelta. Los de 3-1/2 no utilizan este sistema y carecen de dicho orificio.

- Zona de arrastre.

Sirve para centrar el disco y permitir el arrastre por el motor. En los discos de 5-1/4 se pinza la parte central para producir el arrastre (embrague). En los de 3-1/2 el disco es arrastrado a través de una pieza metálica solidaria del mismo.

En origen, la envoltura del *diskette* presenta una muesca lateral. Si la tapamos con una etiqueta no podremos grabar y así protegeremos la información ante una falsa maniobra. En los de 3-1/2 el mismo efecto se consigue deslizando una pequeña corredera situada en la esquina inferior izquierda.

Hay que poner especial cuidado en no estropear la superficie magnética ya que de otra forma perderemos su contenido de forma irrecuperable (es una buena práctica duplicar nuestra información valiosa en *diskettes* separados).

Indicamos algunas de las cosas que NO DEBEMOS HACER:

- Exponer los *diskettes* a focos de calor, polvo o radiaciones magnéticas (por ejemplo al sol o cerca de un imán).

- Doblar el *diskette* o escribir sobre él con un bolígrafo (las etiquetas deben escribirse antes de pegarlas o si no con rotulador de punta blanda).

- Tocar el material magnético con la mano a través de la ranura de lectura / escritura o intentar limpiar el disco con un trapo.

Conviene que puntualicemos algo sobre la terminología. A veces al *diskette* (también veremos escrito disquette o disquete)se le denomina disco de forma genérica. Desgraciadamente también se denomina así al periférico que arrastra el disco y permite su uso (*driver* en inglés) y sólo por el contexto sabremos de qué se está hablando. Para evitar esta confusión se suele decir «disco» y «unidad de disco», pero no está aceptado universalmente.

En los discos musicales la información se guarda en un solo surco o pista colocada en forma de espiral. En los *diskettes* la información se almacena siempre en pistas circulares concéntricas, las cuales a su vez estan subdivididas en sectores (ver figura).

El número de pistas y sectores varía entre marcas y para ello debe acudirse a la hoja de características respectiva. No obstante podemos considerar valores típicos 40 u 80 pistas, 9 sectores y 512 bytes . En **MSX** la configuración es 80 pistas, 9 sectores y 512 bytes por sector , con lo que se alcanza 360 Kbytes por disco. Dependiendo de la marca y el modelo, el *diskette* puede estar preparado para almacenar una cara, aunque las máquinas de la segunda generación empiezan a usar las dos.

Otros parámetros que conviene mencionar al hablar de *diskettes* son :

- Densidad de grabación.

Indica la mayor o menor cantidad de partículas magnéticas por unidad de superficie. Normalmente se clasifican en «densidad sencilla» y «doble densidad» y se expresa en **bpi** (bits per inch = bits por pulgada).

-Densidad de pistas.

Número de pistas por pulgada (tpi)

- Velocidad de acceso.

Tiempo requerido para comenzar a leer o escribir una información. Se expresa en mseg (milisegundos).

- Velocidad de transferencia.

Velocidad de salida / entrada de datos hacia / desde el ordenador. Se expresa en kbits/sec (Kilobits por segundo).

- Consumo.

Los valores típicos son 10 - 25 watios.

El disco es arrastrado a velocidad constante. La cabeza tiene un movimiento tranversal, controlado por el programa, para posicionarse sobre una determinada pista.

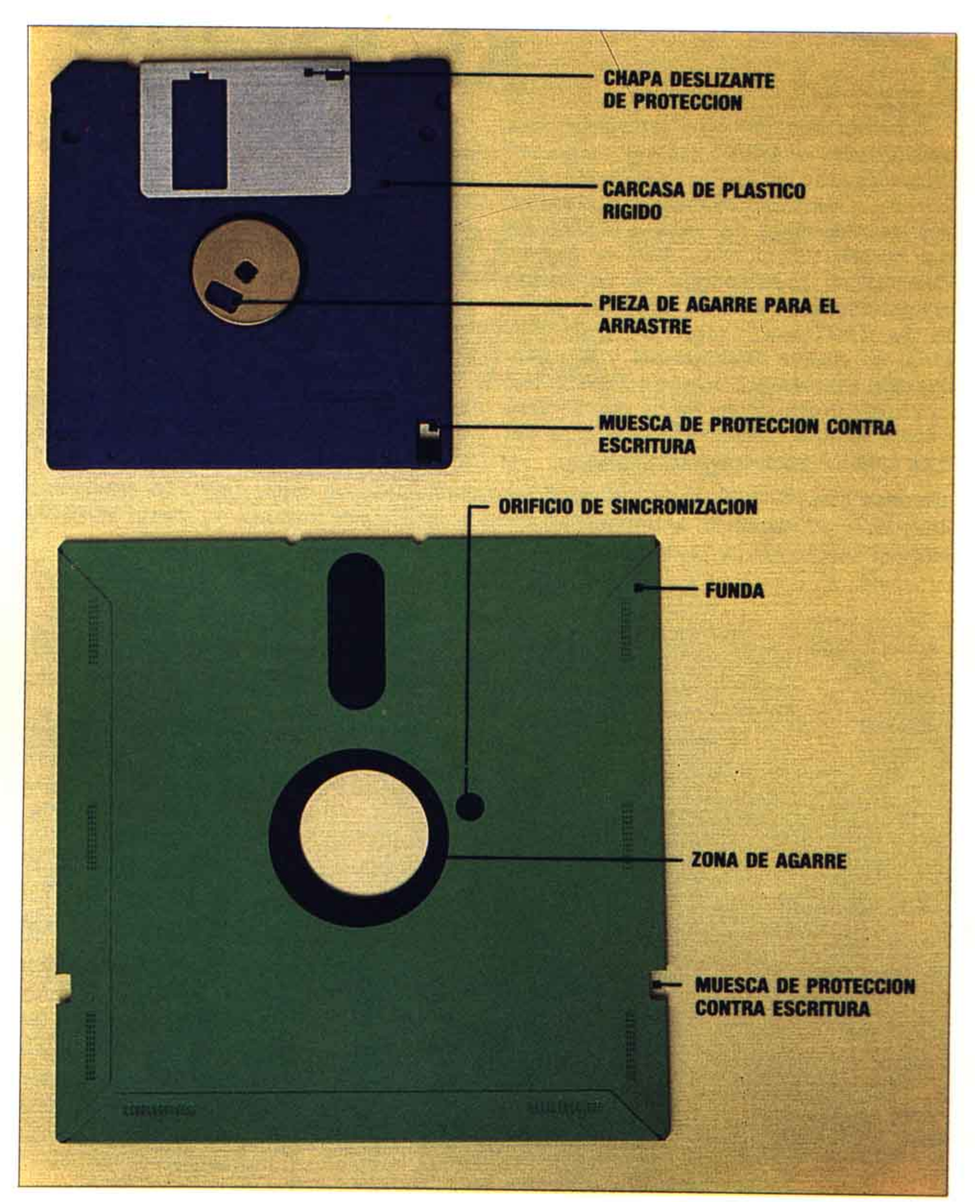
La cabeza es una parte delicada y hay que poner especial cuidado durante el transporte de la unidad de *diskettes*. Es buena práctica dejar siempre metido un disco además de accionar la palanca que bloquea la cabeza para evitar que quede dañada por algún golpe.

Cuando hablamos de pistas y de sectores no queremos decir que el material magnético esté fisicamente agrupado de esta manera. Un disco virgen, tal como viene de fábrica, no sirve para trabajar. Hay que formatearlo con ayuda de un programa especial, de forma que quede dividido en sectores desde el punto de vista del *software*.

Durante el proceso de formateo el programa analiza si hay algún sector defectuoso y toma nota de él en un fichero que guarda en una de las pistas (generalmente la 0, pero no en todas las marcas). En lo sucesivo ignorará estas partes defectuosas. Esta es la razón por la que dos *diskettes* nuevos pueden tener capacidades disponibles ligeramente distintas.

Otra de las grandes diferencias respecto al *cassette* radica en la posibilidad de grabar o leer información de forma aleatoria, además de secuencial. O dicho de otra manera, podemos leer un dato que se encuentre al final de un fichero grande sin necesidad de leer uno a uno los datos anteriores, con lo cual los tiempos de acceso se reducen espectacularmente.

Todos hemos sufrido las incomodidades de buscar un programa en un *cassette* de larga duración donde tenemos guardados otros donde debemos ir guardando datos con frecuencia, seguramente después de algunos tanteos habremos desistido. El *diskette* necesita ayudarse de un programa denominado **DOS** (**Disk Operating Sistem**) y el *interface* correspondiente. Mediante el **DOS** se dispone de nuevos comandos para Borrado de ficheros, Copias a / de otros dicos, Save, Load o Merge de programas, etc. Con ayuda de programas de Utilidades, suministrados generalmente en el mismo *diskette* del **DOS** se dispone de herramientas cómodas de usar para hacer copias de toda o parte de la información de un *diskette* a otro (COPY o BACKUP) y formateado (FORMAT).



El *interface* para disco unas veces viene incorporado en el propio ordenador, otras en la unidad de disco u otras es una unidad separada que se enchufa entre ambas, como es el caso de **MSX**.

Las unidades de *diskette*, al menos las **MSX**, deben llevar su propia alimentación. Existen casos como el del ordenador **BBC** donde la unidad de disco se alimenta del ordenador a través de un enchufe especial. Incluso hay situaciones en las que el ordenador se aprovecha de la fuente de alimentación de la unidad de disco, como es el caso de la **Opus Discovery 1 ZX-Spectrum**.

En su aspecto exterior todas las unidades de disco (*drivers*) son muy parecidas. Disponen de una abertura en forma de ranura por donde introducir el disco y un piloto rojo (**LED**) para indicar si está en reposo o en fase de lectura / escritura. Unas tienen unas palanca que hace de cerradura (el disco no puede trabajar si esta palanca no está bajada y no puede introducirse o sacarse si lo está) y en otros como los del tipo **MSX**, el disco se introduce por presión quedando bloqueado hasta que se oprime una palanca de expulsión.

CONCURSO INPUT/ANAYA MULTIMEDIA

¡DESCUBRE AL ASESINO!

BASES:

Durante tres números sucesivos irán apareciendo en **INPUT** las tres partes del relato «El caso del anciano asesinado» junto con los mensajes cifrados que constituyen las respuestas parciales a la solución del crimen. Cada mes, quienes logréis descifrarlos participaréis en un sorteo consistente en 10 lotes de libros por un valor de 15.000 ptas., a elegir del fondo editorial de **Anaya Multimedia**, y una suscripción gratuita por un año a **INPUT**. Habrá tres sorteos, de tal forma que no será necesario haber descifrado el enigma del mes anterior para optar al premio.

Un fabuloso premio será la guinda final de este pastel. El descubridor de las motivaciones del crimen visitará al próximo **PCW Show** a celebrar en Londres. En caso de haber más de un acertante, recurriremos a la inevitable fórmula de sorteo.

La solución del tercer mensaje cifrado deberéis enviarla, junto con vuestros datos personales, a la Redacción de **INPUT** antes del 15 de agosto.

Las decisiones del jurado serán inapelables, dándose en las páginas de **INPUT** cumplida cuenta de la marcha del concurso.



Quienes deseeis optar al premio especial debeis enviar en sobre aparte las respuestas a todos los mensajes. La fecha límite es también el 15 de agosto. El afortunado recibirá la noticia por teléfono, ya que el PCW Show *se celebra a principios de septiembre. Su nombre aparecerá publicado en INPUT de octubre.*

NOTA:
El plazo para la recepción del segundo mensaje es el próximo 15 de julio, aunque por error apareció la misma fecha que para el primero.



EL CASO DEL ANCIANO ASESINADO

—Me siento culpable —decía entre lágrima y lágrima—. Si yo hubiera estado con él, tal vez no habría muerto.

Intenté consolarle, pero la verdad es que el viejo me producía cierta repugnancia. Anoté:

```
CSQESOXDO MSPBKNY JSQ JKQ DWVOABKSELBKNV
CEKKRMODBOCCYYOZBCOXCONMKCBPXROBOYNBYVMB
ESKNVVMOYVVSKWCCKKKWSQKDC
```

Por lo demás, también vivía solo, como su hermano. Al parecer, los dos eran los últimos miembros de una familia de rancio abolengo, cuya grandeza había entrado en franco declive. De sus antiguas riquezas apenas quedaban ya unos restos: tres cuadros de firmas importantes, una pequeña casa junto al mar que casi no podían mantener y las joyas que habían robado de la caja fuerte de su hermano.

—Esto es el fin —dijo el pobre viejo entre sollozos—, esto es el fin de una importante dinastía y Dios ha permitido que viva para verlo.

Cuando llegué a la comisaría, mi ayudante había salido, pero sobre la mesa de mi despacho me había dejado una nota con el siguiente texto:

```
PFDRFBKQB ZFCOXAL QOXKPMLPFZFLK FQFKBOXO
FL BKIDOBOFLZBPBQAQCRLKBXBIXZXXIZPAXQXBS
IPXRXYBCAXIBXGPIALRLBIALJLWOFOWWQFWWWLWW
WWWWWWW
```

Junto a la nota estaba la colilla que, supuestamente, había originado el incendio y que, sin duda, era la clave del enigma. Una sonrisa se dibujó en mi rostro cuando anoté:

```
RHFTHDMSD BHEQZCN FQNMREDKC D WITGW DI G
VAES HFSXCQAI RAJMEFD LR MG HOPKOSE TK U
OHKDU HJYYIRIXPV QKWFIEWHQFTYE PCV TESIF
S HGMHHBY UOV NRZ MOINSMXRZ HF AS FYODKS
S TJRZKRZS
```

... Había resuelto un nuevo caso.

Juan José Millas



LOS MEJORES DE INPUT MSX

PUESTO	TITULO	PORCENTAJE
1.°	Knight Lore	20,5 %
2.°	Profanation	13,9 %
3.°	Soccer	13,7 %
4.°	Boxing	12,7 %
5.°	Road Fighter	8,9 %
6.°	H.E.R.O	8,8 %
7.°	The way of the tiger	6,9 %
8.°	Alien 8	5,8 %
9.°	Yie ar kung-fu	4,9 %
10.°	Ghostbusters	3,9 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Julio de 1986





SOTANOS

1
2 3 4 5 6
7

1
2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27 28
29 30 31 32 33 34 35 36 37 38 39 40
42 43 44 45 46 47 48 49 50 51 52 53 54
57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74
81 82 83 84 85 86 87
89 90 91
92
93
94
95
96
97

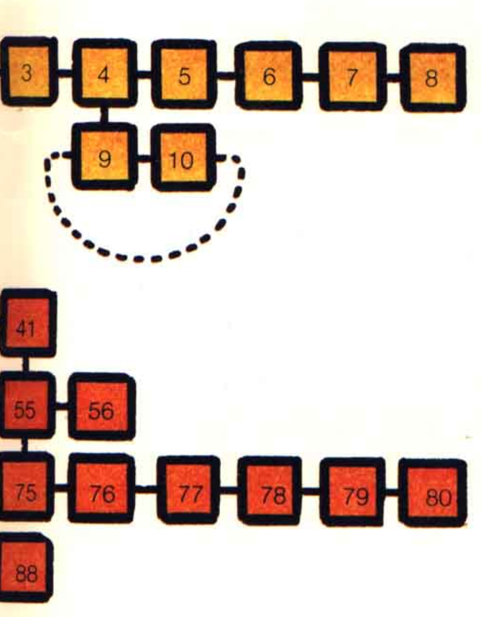
1 2

NAVE ESPACIAL

1
2 3 4 5
6 7 8 9 10 11 12 13 14
15 16 17
18 19 20 21 22 23 24

AL PLANETA

Jet Set Willy II no tiene mucho que ver con lo que normalmente entendemos por una segunda parte. A diferencia de lo que cabría esperar, no se trata de la típica continuación de la aventura, en la que solemos encontrarnos ante nuevas situaciones y distintos objetivos, sino de una simple reedición ampliada del mismo programa original. Gráficos, ambiente, tema ... todo igual, aunque, eso sí, con más pantallas. El juego no deja por ello de ser



excelente en su conjunto, pero estamos seguros de que más de un usuario se va a llevar un chasco al comprobar que le han vendido un programa que es prácticamente igual a otro que ya tiene. No podíamos dejar de hacer la advertencia.

TE LO CONTAMOS

Willy, el simpático personaje que tan buenos ratos nos hizo pasar con el **Jet Set Willy I**, sufrió un grave accidente tras su última aventura: se cayó por las escaleras y estuvo a punto de matarse. Durante su larga estancia en el hospital, tomó la sabia decisión de contratar a un grupo de albañiles para que arreglaran los escalones que provocaron la caída, con el objeto de evitar desgracias mayores. Pero a su vuelta a casa, descubrió horrorizado que los «supuestos» albañiles se habían tomado ciertas libertades en su ausencia: no se contentaron con la sencilla reparación que se les había encargado, y construyeron pacientemente, y sin permiso de nadie, varias decenas de nuevas habitaciones, como si la casa no tuviera bastante con las sesenta originales (estas nuevas habitaciones son precisamente las que se han añadido a la nueva versión). **Willy** no termina de explicarse esta extraña actitud de los albañiles, que por otra parte, se negaron a cobrarle el trabajo. Lo que más le intriga es el hecho de que tuvieran la piel verde, aunque la existencia de una habitación cohete no es menos inquietante.
Finalmente, nuestro héroe se decide a visitar una por una todas las estancias de su enorme casa, con el ánimo de desvelar el misterio, aunque le cueste varias de sus muchas vidas.
María, su ama de llaves, está de acuerdo con la decisión (aunque lo que verdaderamente pretende es que le limpie los escombros) y para terminar de convencerle no duda en utilizar el persuasivo argumento del rodillo de cocina.
Esta novelesca trama es todo lo que hemos podido sacar en claro de las instrucciones que acompañan al juego. Lo que éstas no dicen, y lo que nosotros hemos logrado descubrir, podréis encontrarlo bajo el epígrafe **la voz de su amo**, unos párrafos más adelante.

POR MAS SEÑAS

Jet Set Willy II, en su versión para **MSX**, es un programa creado por **Software Projects Ltd.** Utiliza un sistema de protección diseñado por **Padlock Systems**, basado en una matriz de colores que se corresponde con un código que el usuario debe introducir para acceder al programa.



Por tratarse de un código de colores, el clásico recurso de la fotocopia queda descartado. Lo sentimos por los «piratas».

DATOS GENERALES

TITULO Jet Set Willy II

FABRICANTE Software Projects

CLASE DE PROGRAMA Arcade tipo Manic Miner

FORMATO Cassette

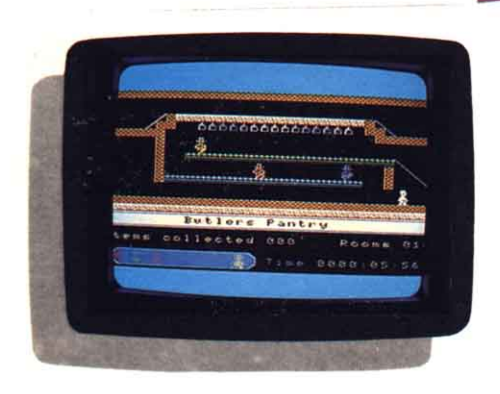
CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	3
INTERES	8
GRAFICOS	6
COLOR	6
SONIDO	5
TOTAL	28

LO BUENO Y LO MALO

Entendemos que un buen programa debe reunir varias condiciones





indispensables para que realmente lo sea, y todas ellas se resumen en una: que nos haga pasar buenos ratos.
Jet Set Willy II reúne esa condición, aunque desgraciadamente presenta algunos aspectos negativos que pueden ensombrecer un poco su valoración general. En primer lugar, debemos reseñar su escasa originalidad y su excesivo parecido con el clásico modelo, de **Arcade**, ya muy superado, tipo **Manic-Miner**. Por otro lado, esa nueva versión no ha incorporado ninguna de las mejoras que necesitaba el programa original, sino que simplemente presenta un aumento en cuanto al número de pantallas, y algunos pequeños cambios referidos a la forma de conseguir el objetivo. Lógicamente, estos detalles pueden decepcionar a aquellos que adquieran esta falsa segunda parte pensando encontrar algo nuevo, como ya advertíamos al principio de este comentario.
A pesar de todo, **Jet Set Willy II** tiene algo. Y ese algo es precisamente lo que define a los buenos programas.



LA VOZ DE SU AMO

En este apartado vamos a tratar de desvelar esos pequeños secretos que tantos quebraderos de cabeza suelen provocar en quienes no tienen a mano el número correspondiente de **INPUT**.
Jet Set Willy II tiene un total de 139 habitaciones. Algunas de ellas comunican con partes aisladas de la casa que no se corresponden directamente entre sí. Por ejemplo, las cuerdas que podréis encontrar en la habitaciones 44 y 61 (ver mapa), conducen respectivamente a las habitaciones 2 y 5 de la zona señalada con el nombre de **Los Sotanos**.
En la habitación número uno del plano principal, hay dos objetos que **Willy** puede recoger. Tomándolos, será transportado instantáneamente (tras esperar unos segundos) a la habitación 21 de la zona **Nave Espacial**. Desde la habitación número 6 de este mismo sector, se accede a una de las estancias de la zona **Planeta**.
Sabemos que no conviene excederse destripando el «intríngulis» del juego, y por ello no vamos a daros más que una pista sobre el último obstáculo que debéis salvar: mucho cuidado con la habitación 91, pues de ahí en adelante cualquier intento de pasar se paga con la muerte instatánea del pobre **Willy**. La verdad es que no conocemos a nadie que haya conseguido ver lo que hay en la habitación 97. Tal vez tú logres ser el primero.

VALORACION

Jet Set Willy II es un programa de aspecto algo anticuado, que nos remonta a los primeros momentos de la breve historia del *soft*, pero a pesar de ello, sorprende por su calidad y por su elevado grado de interés. Es una muestra excelente de lo que se puede llegar a hacer con mucha imaginación y pocos medios. Por otra parte, sus 138 pantallas garantizan que por mucho que se juegue, siempre quedará algo nuevo por descubrir. En resumen, merece la pena.
Esperamos que el mapa y los consejos que hemos dado os sean de mucha ayuda, y os permitan no perder los ánimos ante los primeros fracasos.

★★★★★★★★★★★★★★★★★★★★★★★★★

A 300 KM POR HORA

Quizás lo más novedoso de **Le Mans 2** sea la forma de presentación. La memoria que contiene el juego se encuentra integrada en una tarjeta cuyo grosor es un poco más ancho que el de una tarjeta de crédito normal y corriente. Necesita de un adaptador para pasar de las conexiones de la tarjeta a las estandar del cartucho. Este adaptador se inserta en el lugar reservado al cartucho y la tarjeta se inserta en el adaptador.
Son claras las ventajas de este método en cuanto a almacenamiento, transportabilidad, intercambio, etc.
En cuanto al juego en sí, cabe decir que no aporta ninguna novedad interesante en comparación con otros juegos del mismo tipo. Se trata del clásico juego de carreras de coches en el que el jugador siente el efecto de ir en la cabina del coche, observando el circuito en perspectiva



DATOS GENERALES

TITULO Le Mans 2

FABRICANTE Serma

CLASE DE PROGRAMA Juego

FORMATO Tarjeta

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	8
COLOR	8
SONIDO	8
TOTAL	38





real de tres dimensiones.
El efecto de realismo está muy bien logrado, incorporando incluso detalles tales como los brazos del piloto, que giran el volante en un sentido o en otro cuando hacemos maniobrar el coche, e incluso los espejos retrovisores, que «reflejan» la imagen del coche que va a adelantarnos o la del que acabamos de adelantar.
No debemos preocuparnos por el cambio de marcha, ya que es automático. La carrera es a 10 vueltas, con un límite de tiempo de 24 minutos, y en ella competiremos con otros 19 veloces coches. ¡Ojo al combustible y a los posibles derrapes!
Para dar más impresión de realismo a la carrera podremos pilotar nuestro vehículo de noche, como en las auténticas 24 Horas de **Le Mans**.
Con nuestra suerte y habilidad no cabe duda de que llegaremos los primeros a la meta.

★★★★★★★★★★★★★★★★★★★★★★★★

LA NOCHE DE LOS ESPECTROS

¿Conseguirá nuestro héroe rescatar a su chica de las malvadas garras de los muertos vivientes? Sólo depende de su habilidad y rapidez para sortear los múltiples peligros del bosque, del cine, del *ghetto*, del cementerio y de la casa encantada.
Se encuentra solo ante los malvados espectros que intentarán arrebatarle su energía vital, y así evitar que logre hacerse con las cruces salvadoras.
Para recuperar parte de su energía, sólo dispone del alimento que le proporcionan las setas silvestres pero ¡cuidado con las venenosas! Estas últimas no sólo no nos darán energía, sino que nos arrebatarán parte de la que tenemos.
El objetivo del juego es evitar los múltiples peligros y llegar hasta la casa encantada en medio del bosque.
Allí, nuestro héroe deberá rescatar a la chica y escapar hacia el coche. En la huída, la salvación dependerá de







DATOS GENERALES

TITULO Chiller

FABRICANTE Mastertronic

CLASE DE PROGRAMA
Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	9
GRAFICOS	8
COLOR	9
SONIDO	7
TOTAL	42



ambos, ya que en el camino hay cruces que sólo puede tomar el chico y cruces que sólo puede tomar la chica.
No dudamos de que con vuestra ayuda el rescate se llevará a cabo felizmente y de que, además, podréis pasar ratos muy divertidos con este original juego. Ah..., y ¡cuidado con los espectros!

****

EL PROFANADOR DE TEMPLOS

Johnny Jones intenta salir con vida del oscuro y misterioso templo de **Abu-Simbel**. ¡Seguro que con nuestra ayuda lo logra!
Nuestro héroe se encuentra perdido en los tortuosos pasadizos y siniestras galerías. Su ansia de aventura y su afán de riqueza le hicieron profanar el templo sagrado y ahora se encuentra bajo los terribles efectos del hechizo de **Ramses II**, que ha hecho desaparecer parte de su cuerpo.
Para recuperarlo y lograr salir con su cuerpo completo y con las riquezas del templo deberá llegar a la cámara mortuoria en la sala de la esfinge. Pero no es tarea fácil. Los pasadizos del templo se encuentran llenos de trampas, líquidos corrosivos, voraces pirañas, serpientes, arañas gigantes, etc. **Johnny** tendrá que sortearlos todos y encontrar el camino adecuado dentro del laberinto de

DATOS GENERALES

TITULO Abu-Simbel

FABRICANTE Dinamic

CLASE DE PROGRAMA
Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	8
GRAFICOS	8
COLOR	8
SONIDO	7
TOTAL	38

pasillos. En algunos de ellos incluso tendrá que arreglárselas para descubrir el modo de abrir el pasadizo que conduce al camino correcto.
Por fin, tras resolver el enigma del diamante, será conducido a los interiores del templo, desde donde podrá llegar a la cámara mortuoria, tras atravesar otra serie de terribles trampas. Allí descubrirá el secreto del templo.
Es realmente muy entretenido ayudar a **Johnny** a completar su aventura. **Abu-Simbel** se encuadra dentro del conjunto de juegos que podrían llamarse de laberinto de peligros en los cuales un personaje debe moverse dentro de una serie de pasadizos y llegar a una meta venciendo un montón de dificultades. Como todo este tipo de juegos tiene sus ventajas y sus inconvenientes. Puede llegar a ser muy ameno, pero una vez nos aprendemos lo que hay que hacer para pasar las trampas, nos divertimos menos.
Los efectos sonoros consisten simplemente en una música de fondo (graduable en intensidad desde el teclado, lo cual es un acierto pues evita que al final la música se termine haciendo un poco pesada).
El movimiento del personaje se limita a desplazamientos a derecha e izquierda y a saltos, lo cual es sencillo pero desde nuestro punto de vista un poco pobre.
Disponemos de un total de 16 vidas de **Johnny Jones**, para que éste pueda llegar a su objetivo. Son más que suficientes cuando se tiene un poco de experiencia en el juego, pero es una gran idea, ya que evita los tediosos comienzos desde el principio cuando ya se está en una fase avanzada del juego.




LIO EN EL RIO

Nos encontramos en plena misión aérea de combate. Nuestro objetivo es destruir el mayor número posible de barcos e instalaciones enemigas, a lo largo del rio, con nuestro reactor. Pero cuidado, ya que, a medida que vayamos remontando el caudal, el número y la calidad de nuestros enemigos aumentará, así como la sinuosidad del río, que se irá haciendo más estrecho y tortuoso. Podemos guiar el avión con la palanca del *joystick* o las teclas del cursor, para hacerlo girar a derecha e izquierda, y acelerar o decelerar. ¡Ojo con el combustible! Se irá consumiendo regularmente a medida que avanzamos, y llegará un momento en el que nos veremos obligados a repostar. Entonces, tendremos que acercarnos a la velocidad adecuada a los tanques de combustible. El indicador de fuel emitirá un sonido cuando el tanque esté a 1/4 de su capacidad, avisándonos de la necesidad de repostar. Durante el proceso de carga, un sonido más agudo nos avisará del llenado del tanque. ¡Cuidado con repostar más de lo necesario, pues el tanque estallará! Al comenzar el juego podemos escoger el nivel de dificultad, situándonos en puestos de comienzo más avanzados a lo largo del curso del rio (y por tanto dejándo atrás trayectos más fáciles), cuando creamos que nuestra experiencia y pericia lo merece.
Los efectos visuales no son especialmente espectaculares, por lo menos en comparación con el estado actual de este tipo de juegos. Se limitan a una visión «a vuelo de pájaro» de nuestro avión, del río y de los dispositivos enemigos.
Los efectos de movimiento del avión no son tampoco muy vistosos. El juego, sin embargo, resulta en su conjunto entretenido.

DATOS GENERALES

TITULO River Raid

FABRICANTE Activision

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	7
COLOR	8
SONIDO	8
TOTAL	37







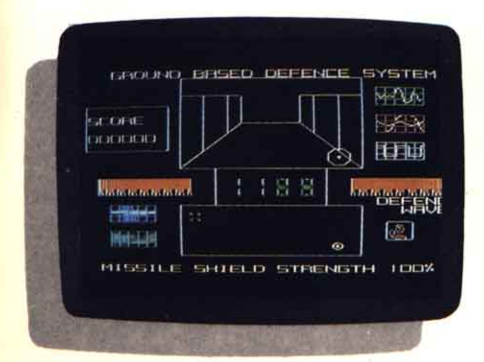
★★★★★★★★★★★★★★★★★★★★★★★★★★★★

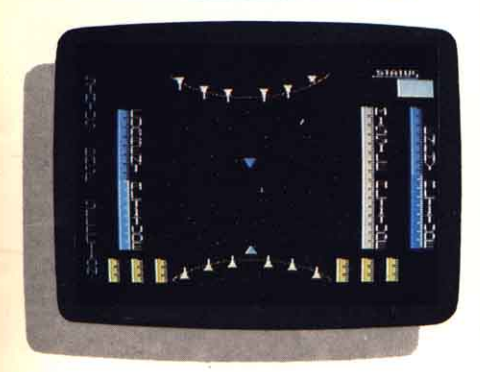
EL KAMIKAZE

¡Atención Base! ¡El avión enemigo se acerca! Tenemos que salir en nuestro reactor **Crusader** a enfrentarnos con él en un fabuloso combate combinando aire-aire y tierra-aire.
Nuestro objetivo es destruir el avión enemigo y después dirigirnos con nuestro caza a la base de tierra enemiga, para destruirla también con nuestros misiles.
Para ello gobernaremos nuestro reactor **Crusader** y dispondremos además de una base de misiles de apoyo en tierra, que puede servirnos de ayuda en caso necesario.
El combate se desarrolla en dos fases. Primero tendremos que ir al encuentro del avión enemigo para luchar con él y derrotarlo. En el trayecto tendremos también que destruir los satélites enemigos ya que si logran colisionar con nosotros debilitarán nuestro escudo protector.
En cualquier momento podemos solicitar ayuda de nuestra base terrestre. Pasaremos entonces a



controlar el disparo de los misiles desde tierra. Seleccionando adecuadamente su altitud y su





dirección los haremos llegar hasta las inmediaciones del avión enemigo. Una vez allí, ellos mismos se guiarán por el calor que desprende, hasta alcanzarlo.
Del resultado del combate aéreo dependerá la siguiente fase del juego. Si ha sido favorable, y hemos logrado destruir el avión enemigo, pasaremos a atacar su base, disparando un misil que tendremos que guiar para hacerle esquivar las defensas, hasta hacerlo llegar a la



DATOS GENERALES

TITULO Jet Fighter

FABRICANTE Aakosoft

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	10
GRAFICOS	9
COLOR	9
SONIDO	9
TOTAL	46

puerta de la sala de control, donde estallará.
Si el resultado ha sido desfavorable, saltaremos en paracaídas desde nuestro avión en llamas, para ir rápidamente a defender nuestra base en tierra. Intentaremos destruir el misil lanzado por el enemigo, disparando desde nuestras barreras defensivas. No es fácil, pues el cohete enemigo está atiborrado de dispositivos electrónicos para detectar y esquivar nuestros disparos. Ganará el juego el primero que consiga destruir la base enemiga tres veces.
Nuestro **Crusader** no tiene nada que envidiar a los aviones de verdad. Dispone de una completa cabina de control con un cuadro de mandos que indica el nivel de combustible, la posición del caza enemigo, la dirección en la que se encuentra la base enemiga, la velocidad horizontal



y vertical, la altura, temperaturas del motor y del laser y un ordenador de a bordo que nos avisará de diversas condiciones de alarma.
Jet Fighter es una verdadera obra de arte de los gráficos y también del sonido, ya que incluso ¡habla!. Os lo recomendamos de veras si queréis pasar ratos verdaderamente entretenidos.

****

BATALLA DE REYES

No podían tardar mucho en aparecer los juegos de ajedrez para **MSX**, lo mismo que ya existen para otros sistemas y marcas de ordenadores.
En esta ocasión comentamos **Ultra Chess**, un buen juego de ajedrez con una gran variedad de opciones que pasamos a analizar.
Cuando se termina el proceso de carga, el ordeandor nos da a elegir entre tres posibilidades: «P» para jugar una partida, «A» para entrar en el proceso de análisis, o «L» para cargar un juego que hubiéramos salvado anteriormente en *cassette*.
En la opción de jugar, primero escogeremos piezas (blancas o negras) y fijaremos el tiempo límite que tendrá el ordenador para pensar su jugada. Comenzará entonces el juego. En la parte derecha de la pantalla aparecerá el tablero con la disposición de las piezas durante la partida, y en la parte izquierda se



nos mostrarán los ocho últimos movimientos realizados, así como los cronómetros medidores del tiempo por jugada.

EL SILBIDO DE LAS BOMBAS

DATOS GENERALES

TITULO Jet Bomber

FABRICANTE Aackosoft

CLASE DE PROGRAMA Combate aéreo

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)	
ORIGINALIDAD	8
INTERES	8
GRAFICOS	7
COLOR	6
SONIDO	7
TOTAL	**36**

La trama de **Jet Bomber** se desarrolla en la **República Unida de Abadeer** (una de las muchas repúblicas petrolíferas de Oriente Medio). **Hassan el Inclemente**, líder del frente de liberación de **Abadeer**, ha iniciado una campaña de atentados terrorístas destinada a entorpecer las actividades de la compañía petrolífera **Aceites Oxxan**, que opera en la República. Ante este ataque, la dirección de la compañía petrolífera ha optado por la solución más radical: contratar los servicios de una red de mercenarios, que dispone de un fabuloso bombardero. El objetivo es el de acabar de una vez por todas con el molesto **Hassan**. Aquí es donde entra el jugador, que hará las veces de piloto mercenario. A los mandos del bombadero y empleando a fondo sus recursos, destreza y nervios de acero, tiene la misión de internarse en territorio rebelde para bombardear y destruir todos los objetivos que encuentre.





El bombardeo tendrá lugar en un escenario gráfico tridimensional de bastante calidad (recuerda por su aspecto al de un conocido juego que lleva el nombre de **Zaxon**).
El movimiento, *scrolling* y velocidad del juego son también excelentes. A todo ello hay que añadir unas cuantas frases digitalizadas que se pueden escuchar por el altavoz del televisor en los momentos cruciales del juego. Con todos estos elementos, **Jet Bomber** merece la calificación de muy bueno.

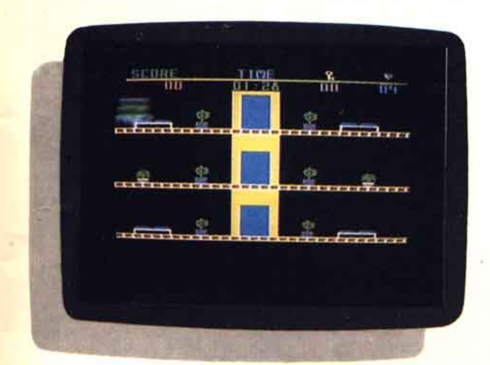
★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

EL ESPIA DEL MUSEO

¡Unete al superespía **Graham Crackers**, en otra de sus magníficas aventuras! Tienes que buscar el microfilm oculto tras uno de los cuadros del museo, y escapar con él para entregarlo en tu cuartel general. Si no lo consigues, el enemigo desvelará los terribles secretos que encierra, y se apoderará del mundo. Sin embargo el museo se encuentra en manos de una terrible organización de espías, que ha llenado todas sus salas de trampas y peligros que tu y nuestro superagente deberéis sortear, si queréis llevar a cabo vuestra difícil misión. Moverse por las 90 salas diferentes no va a ser tarea fácil, y menos aún al ir mirando detrás de cada cuadro, en busca del que tiene los documentos secretos, pero seguro que con tu ayuda, **Graham** lo consigue.
Hay peligrosos robots, a los que habrá que esquivar, trampas de



electricidad, rayos mortales y puertas secretas que sólo podremos abrir localizando todas las llaves escondidas.
En cada habitación, podemos movernos en tres pisos. Hay escaleras y ascensores para pasar de uno a otro pero ¡cuidado! no todo es lo que parece y podemos encontrarnos con desagradables sorpresas preparadas por el enemigo.
El juego es sobre todo variado, dado el elevado número de habitaciones diferentes por las que **Graham** puede deambular en busca del microfilm. Sin embargo, tanto los gráficos como el movimiento del personaje se podrían haber mejorado bastante, para llegar a la calidad de muchos juegos parecidos.



DATOS GENERALES

TITULO The Heist

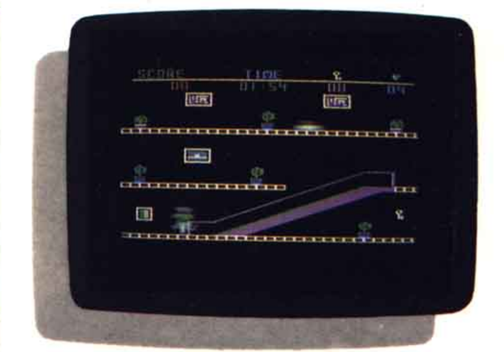
FABRICANTE Aackosoft

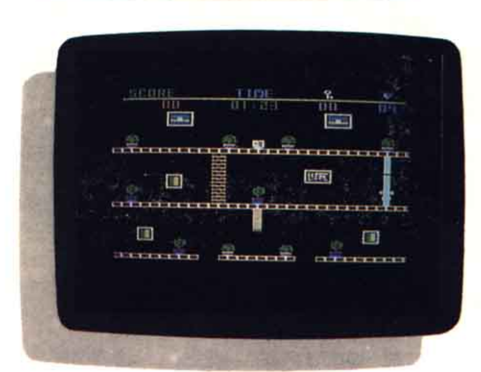
CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	8
GRAFICOS	7
COLOR	8
SONIDO	6
TOTAL	37





GRAND PRIX

¡Participemos con nuestro bólido **MSX** en el campeonato del mundo de Fórmula 1!. Podremos competir en las carreras que se celebran en los 10 circuitos más famosos de todo el mundo: **Silverstone**, **Brands Hatch**, **Mónaco**, **Hockenheim**, **Osterreichring**, **Kyalmi**, **Zolder**, **Paul Ricard**, **Monza** y **Zandroort**.
Montemos en nuestro coche, seleccionemos el circuito que más nos guste y ¡a correr! Disponemos de la posibilidad de elegir entre cambio de marcha automático y manual. La selección de cambio automático, si bien es menos emocionante, es más conveniente, al menos en un principio, sobre todo cuando todavía somos inexpertos en el manejo de nuestro estupendo Fórmula 1. Nos saldremos menos veces de la pista.
No estamos exentos de las inclemencias del tiempo. Si está lloviendo, nos encontraremos la pista mojada, y tendremos que conducir con más precaución.
Fórmula 1 es el clásico juego de simulación de carreras de coches en el que se tiene la sensación de ir dentro del vehículo. No tiene ninguna aportación que le haga especialmente destacable sobre otros juegos de su mismo tipo.

DATOS GENERALES

TITULO Formula 1 Simulator

FABRICANTE Mastertronic

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	6
INTERES	6
GRAFICOS	7
COLOR	7
SONIDO	7
TOTAL	33





Dentro de la partida podremos pedir ayuda, para que el ordenador nos recomiende un movimiento, parar el juego y comenzar otro (siempre nos queda ese recurso si vemos que vamos perdiendo). También podremos retroceder al movimiento anterior, grabar en *cassette* la partida en curso (sin interrumpirla) o imprimirla, así como cambiar los colores del tablero y las piezas.



DATOS GENERALES

TITULO Ultra Chess

FABRICANTE Aakosoft

CLASE DE PROGRAMA Juego de ajedrez

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	8
GRAFICOS	7
COLOR	7
SONIDO	6
TOTAL	35

En la opción análisis, editaremos nuestro tablero de ajedrez para colocar la disposición de piezas que deseemos o bien para modificar la partida en curso. Podremos eliminar todas las piezas del tablero, y colocar las que queramos en las nuevas posiciones, o bien eliminar solamente algunas piezas determinadas de la partida. En cualquier momento tendremos la posibilidad de deshacer los cambios efectuados.
Hay que resaltar que el ordenador es sumamente astuto, siempre que le demos suficiente tiempo para pensar, y seguro que nos ganará más de una vez. Pero no importa, con **Ultra Chess** acabaremos convirtiéndonos en auténticos maestros de ajedrez, preparados para enfrentarnos a cualquier rival, por experimentado que sea.
Es sólo cuestión de analizar muchas jugadas.

★★★★★★★★★★★★★★★★★★★★★★★★

DUELO EN EL HYPERESPACIO

¡Por fin hemos conseguido el secreto del hiperespacio! Pero otros seres lo han logrado antes que nosotros, e intentan monopolizar su uso.
Nuestra nave deberá defenderse de sus ataques hostiles, y cabalgar sobre los rayos de luz del hiperespacio para llegar a su destino.
Sólo podemos confiar en nuestra pericia como experimentados pilotos de naves espaciales, para saltar de rayo en rayo de luz, esquivando los continuos ataques de las naves enemigas, que intentarán destruirnos con sus láser y sus cohetes suicidas.
Pero no estamos indefensos.
Disponemos de un potente láser y de tres torpedos terriblemente destructores. Estos torpedos hay que dosificarlos bien, usándolos únicamente en casos de extrema necesidad, ya que son nuestra única defensa contra algunas naves enemigas a las que el láser no afecta.
Cuando hayamos destruido todas las naves de una pantalla o sector pasaremos al sector siguiente. A medida que vayamos avanzando en pantallas, los sectores que tendremos que explorar estarán más llenos de peligros.
Son realmente espectaculares los efectos gráficos de la retícula en tres dimensiones en la que se desarrolla el juego, así como del movimiento de las naves en dicha retícula de luz.



DATOS GENERALES

TITULO Beam Rider

FABRICANTE Activision

CLASE DE PROGRAMA Juego

FORMATO Cassette

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	9
GRAFICOS	9
COLOR	9
SONIDO	8
TOTAL	44







EL ZOCO

Vendo ordenador Sony HB 55P con ampliación de 16Kb. más cassette Sony TCM2 con remote más televisor Philips 14 pulgadas b/n todo con un año por 60.000 ptas.

Pedro Terradellas Puig
Calella, 8, Bl. 80 1ª 2º
Tel. 395 03 87 (tardes de 3 a 7)
Badalona (Barcelona)

Vendo o cambio Decathlon-MSX, Ghostbuster-MSX, Pit Fall II- MSX, Grand National-MSX.

Sánchez Barcaiztegui, 16
Tel. 433 47 71

Intercambio programas MSX, primeros títulos del mercado.

Iñaki Fernández Izquierdo
Zamakoa, 7, 5º dcha.
Tel. (94) 456 33 72
Galdacano (Vizcaya)

Intercambio programas, ideas y experiencias con el MSX. Tengo entre otros títulos: Hero, Night Flight, Ninja, Facturas, etc.

Francisco Escuder Roberto
Mallorca, 5, Pª 4
Tel. (96) 150 26 96
Alacuas (Valencia)

Deseo cambiar programas o juegos de MSX, con personas que dispongan de este sistema.

Alex Hoyos Güel
Dr. Roux, 47, 1ª 1ª
08017 Barcelona

Intercambio programas MSX.

Oscar Prado Olea
Labróstegui, 7, 5ºE
Tel. (94) 499 55 52
Cruces-Baracaldo (Vizcaya)

Vendo ordenador Yashica 64K sistema MSX, con monitor Philips fósforo verde, cassette Philips tipo consola especial para ordenador, dos joysticks Sony, dos cartuchos, montones de juegos y revistas por sólo 80.000 ptas. Con menos de tres meses de uso. Información llamando a:

Carlos
Tel. 858 17 31
Madrid

Vendo ordenador SVI 318 en garantía, con data cassette 904, manual, joystick, varios juegos. Por 33.000 ptas.

Tel. 675 47 21
Madrid

Contacto con usuarios MSX para intercambio de programas, ideas, etc... Interesados escribir a:

Salvador Cornudella Lanza
Pujos, 80 S.Atico 2ª
Hospitalet de Llobregat
(Barcelona)

Vendo ordenador Spectravideo SVI 728 80K en MSX. Sólo unos meses de uso. Dispongo de cassette y varios programas. Vendería por 42.000 ptas. Llamar a:

Josep
Tel. (93) 725 80 37 (tardes)
ó 711 63 84 (noches)

Cambio programas MSX: Hero, Manic, Rollerball, Booga.

Gregorio A. Sobrá Hidalgo
Murcia, 3
Tel. 25 31 29
14010 Córdoba

Desearía correspondencia con usuarios de MSX. Me interesa todo tipo de información sobre el sonido en el MSX.

Manuel Elías
Claudio Guerin, 8, 9C
41005 Sevilla

Compro cartucho de expansión de memoria 16K o 64K cuyo valor no supere las 4.000 ptas.

Luis Santiago
Trajano, 5, 5ºF
Tel. 20 57 77 (horas de comida)
Granada

Cambio procesador de textos COMPOR (original) por desemsamblador.

Gregorio A. Sobrá Hidalgo
Murcia, 3
Tel. 25 31 29
14010 Córdoba

Desearía contactar con usarios de ordenadores MSX de Madrid y más concretamente de la Alameda de Osuna, para intercambio de programas, ideas y trucos de programación.

José María Gallardo
Galera, 41
28042 Madrid

Intercambio programas con usuarios del sistema MSX sin fin económico. Interesados llamar o escribir a:

Joan Taulats Vila
Ausías March, 65, 3º, 1º
Tel. (93) 245 99 84
08010 Barcelona

Intercambio programas con usuarios MSX e información diversa. Vendo programas de Spectrum por pasarme a MSX.

Jordi Roch Puig
Apartado 274
Vic (Barcelona)

Me gustaría intercambiar programas e ideas con usuarios de MSX, de la zona de Bilbao a poder ser. Los que quieran tener información llamar a:

Joseba
Tel. (94) 435 11 37 (de 20h. en adelante)

Vendo impresora Brother EP-22 especialmente para usarla con Commodore 64 y Spectrum entre otras marcas de ordenadores. El precio sería a convenir.

César Toquero López
Santucho, 73 4C
Tel. 433 68 24
48006 Bilbao

Intercambio programas con usuarios MSX a ser posible de mi localidad. Poseo los siguientes: The way of the tiger, Knight Lore, Zaxxon, Buck Rogers, Gyro Adventure, etc. Dirigirse a:

José M. Ramírez Medina
Espartero, 14, 2ºC
Tel. 26 57 32
Las Palmas de Gran Canaria

Vendo/intercambio programas MSX, grandes títulos (Knight Lore, Nightshade, Zakil Wood, H.E.R.O., Ghostbusters, Alien 8, Zaxxon) y muchos más. Escribir a:

Ivan Triadó
Martorell, 59, 2º, 4ª
Sant Cugat (Barcelona)

Compro unidad de disco MSX nuevo a precio de (oferta especial). Compro programa de facturación MSX (diskette). Compro cable unión micro MSX/impresora Seikosha GP-55OA.

Alberto García Noguera
Cantante Merce Melo, 1, 2º 6ª
Gandía (Valencia)

Intercambio juegos.

Teófilo Hernández
Tel. (965) 80 63 26
Ricardo Pérez
Tel. (965) 80 02 82



MSX: GRAFICOS Y SONIDO

Autor: Rainer Laüen
Editor: Ferre Moret
Páginas: 463
Precio: 2.800 ptas.

A cualquier poseedor de un ordenador **MSX** no pueden escapársele las estupendas herramientas gráficas y de sonido que están a su disposición. Es más, seguro que cuando vemos algunos de los estupendos efectos de color, animación y música, de los juegos que hay en el mercado, nos preguntamos: ¿Cómo pueden conseguirse? ¿Podré hacerlo yo alguna vez?

MSX Gráficos y Sonido nos da una respuesta clara a esas preguntas. Es un estupendo libro, que nos ayudará a convertirnos en verdaderos expertos en la programación de efectos sonoros y gráficos.

El libro se divide en dos partes fundamentales. En la primera parte se da una explicación amena de todos los comandos del BASIC **MSX** que tienen relación con ambos temas. Estos comandos se encuentran agrupados de forma lógica atendiendo a sus objetivos. Su presentación es muy variada y distraida y al final de cada capítulo se incluyen pequeños programitas sencillos de aplicación, que nos ayudan a entender claramente el efecto de los comandos explicados.

En la segunda parte se ofrecen unos programas interesantes de aplicación, incluyendo dos juegos: **Fretty** y **Mr. Minner**, que pueden servir de base al lector para aprender los fundamentos y usos de los gráficos y el sonido en los juegos, ayudándole así considerablemente a la hora de crear los propios. Hay además unas rutinas en lenguaje máquina enfocadas a que el lector descubra la importancia de esta herramienta a la hora de acelerar a los programas. El libro finaliza con un sumario de las direcciones de más utilidad en la gestión de los efectos sonoros y gráficos en el **MSX**.

Nos encontramos sin duda frente a un libro que nos ayudará considerablemente a comprender el uso de estas dos potentes utilidades. Sólo cabe achacarle un defecto: en el uso de estas herramientas, primordialmente orientadas a la gestión de juegos, es sumamente importante hacer hincapié en la programación en código máquina, y este detalle creemos que ha quedado incompleto en el libro, por lo demás sumamente útil y entretenido.

CODIGOS Y CLAVES SECRETAS

Autor: Gareth Greenwood
Editor: Anaya
Páginas: 308
Precio: 1.378 ptas.

No es preciso ser un avezado agente secreto, curtido en las redes del espionaje internacional, para sentir cierta curiosidad por la codificación y el desciframiento de mensajes. Sin embargo, hasta ahora era preciso disponer de cierta formación para tener entrada en esta compleja y apasionante disciplina. En adelante, este libro ofrece multitud de ideas y pautas, capaces de iniciar a cualquier usuario de ordenador en el apasionante juego de complicar la existencia de los intrusos de la información o, por otro lado, desgranar los misterios ocultos tras una nota cifrada. La más importante ventaja de la obra consiste en poner la criptografía al alcance del BASIC.



Se describen métodos tan simples como la clásica permutación de las letras del alfabeto, hasta los diferentes tipos de transposición, sin olvidar los cifrados **Gronsfeld**, **Bifido** o **Hill**.

Un loable apéndice se ocupa de las peculiaridades ofrecidas por nuestro idioma, particularización clave en los métodos que recurren al análisis de la frecuencia de la presencia de las letras, que presentan un perfil diferente del idioma en que fue escrito el libro.

La profusión de listados de impresora convierte a **Códigos y Claves Secretas** en una herramienta práctica, que huye de la agobiante teoría.

Por último una nota, aunque se utiliza un BASIC bastante estandarizado, los listados han sido adaptados para ser tecleados en buen número de ordenadores de gran difusión. Este es otro ejemplo del buen hacer de la colección **Anaya Multimedia**.

DESPEGA CON TU MSX

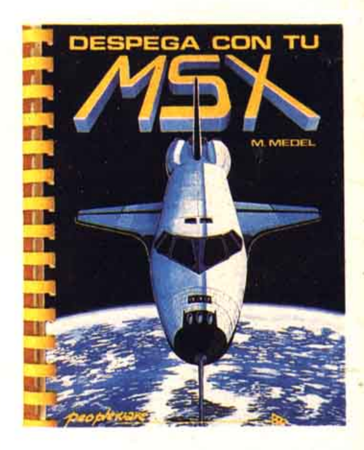
Autor: M. Medel
Editor: Peopleware
Páginas: 241
Precio: 2.300 ptas.

Nos parece realmente original la forma que tiene el autor de enseñar a los lectores las peculiaridades del sistema operativo y del BASIC **MSX**.

En lugar de presentarnos otro manual de **MSX** en la forma convencional (presentación de cada comando y explicación de su función, seguida de uno o varios ejemplos sencillos), ha preferido explicar el uso de todos los comandos a través del desarrollo de un programa concreto.

Dicho programa emula las distintas fases por las que pasa la nave espacial **Columbia** desde su lanzamiento a su aterrizaje. Poco a poco iremos confeccionando el programa partiendo de los bloques más generales hasta llegar a las sentencias concretas, y al final veremos a la nave despegar desde la rampa de lanzamiento que nosotros mismos habremos diseñado, desacoplar los cohetes auxiliares, ponerse en órbita alrededor de la Tierra y proceder a la difícil misión de rescatar un satélite. Y todo ello sin ningún conocimiento previo.

El libro comienza con una breve explicación del estandar **MSX**, y unos someros conceptos de programación estructurada, para después pasar a la



descripción de los comandos a medida que se va realizando el programa.

El contenido se desarrolla de manera muy lógica, partiendo de los bloques principales para pasar después a niveles mayores de detalle.

En la explicación de los comandos no se sigue ningún orden preestablecido, sino que se van introduciendo a medida que son necesarios en cada una de las fases del desarrollo. Ello aumenta en gran medida el interés del lector que, lejos de aburrirse con una descripción tediosa de los comandos, los va aprendiendo a medida que los usa.

El libro nos presenta una nueva metodología para la enseñanza del **MSX** a aquellas personas que no tengan conocimientos previos acerca del estándar.

Konami SHOP®

La marca nº1 en videojuegos para ordenador abre su primera tienda de Europa en Madrid



TODOS LOS JUEGOS DEL CATALOGO DE SERMA ESTARAN TAMBIEN A LA VENTA EN: Konami® LOS TITULOS EDITADOS POR Konami® PARA SPECTRUM, CM 64, AMSTRAD Y MSX SON: FABRICADOS Y DISTRIBUIDOS EN EXCLUSIVA POR SERMA.

Rogamos a todos los usuarios de Spectrum, CM 64, Amstrad y MSX que nos escriban a SERMA sugiriendo ideas para hacerlas realidad en nuestra tienda